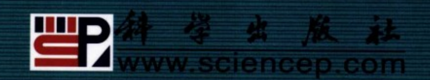
现代物理基础丛书



实验数据多元统计分析

朱永生　编著



现代物理基础丛书　24

实验数据多元统计分析

朱永生　编著

科学出版社

北　京

内 容 简 介

本书介绍实验或测量数据的多元统计分析方法，内容包括：贝叶斯决策、线性判别方法、决策树判别、人工神经网络、近邻法、概率密度估计量法、***H*** 矩阵判别、函数判别分析、支持向量机法等，以及不同判别方法的比较. 此外，还简要介绍了将多种多元统计分析方法的计算机程序汇集在一起的程序包 TMVA(toolkit for multivariate data analysis)，并分析了粒子物理实验数据分析中应用多元统计分析方法的一些实例.

本书可供实验物理工作者和大专院校相关专业师生、理论物理研究人员、工程技术人员及从事自然科学和社会科学的数据测量和分析研究人员参考.

图书在版编目(CIP)数据

实验数据多元统计分析/朱永生编著. —北京：科学出版社，2009

(现代物理基础丛书；24)

ISBN 978-7-03-023676-0

I.实… II.朱… III.实验数据-多元分析：统计分析 IV.O212.4

中国版本图书馆 CIP 数据核字 (2009) 第 016398 号

责任编辑：胡 凯 张 静／责任校对：陈玉凤

责任印制：钱玉芬／封面设计：王 浩

科学出版社 出版

北京东黄城根北街 16 号

邮政编码：100717

http://www.sciencep.com

铭浩彩色印装有限公司 印刷

科学出版社发行 各地新华书店经销

*

2009 年 2 月第 一 版 开本：B5(720×1000)

2009 年 2 月第一次印刷 印张：12 1/2

印数：1—3 000 字数：237 000

定价：48.00 元

(如有印装质量问题，我社负责调换〈明辉〉)

前　言

复杂大系统的科学研究往往都需要收集和处理大量反映系统特征和运行状态的数据信息, 这类原始数据集合由于样本数量巨大, 刻画系统特征的指标变量众多, 并且带有随机性质, 以致于形成了规模宏大、复杂难辨的数据海洋. 利用统计学和数学方法对多维复杂数据集合进行科学的分析, 挖掘出隐藏在复杂海量数据中的规律和信息, 就是多元统计分析研究的基本内容.

大型高能物理实验就是典型的复杂大系统的科学研究工作. 20 世纪 80 年代末北京正负电子对撞机 (BEPC) 和北京谱仪 (BES) 的建成, 是中国高能加速器实验物理的真正开端. 在北京谱仪上进行实验工作的研究组是以谱仪的名称 (Beijing Spectrometer) 命名的, 简称 BES 合作组, 它是由多国物理学家组成的国际合作研究组, 我国物理学家在其中占有主导性的地位. 北京谱仪成功地运行到 2004 年, 获取了 τ-粲能区海量的高能物理实验数据. 在此基础上, 应用多元统计分析方法对实验数据进行分析, 获得了大量居于当时世界领先水平的物理成果. 其中, τ 轻子质量的精确测量、2~5GeV 能区 R 值的精确测量、共振态 $X(1835)$ 的实验观察、σ 粒子的实验确定, 更是引起当时国际高能物理界广泛瞩目的重大成就.

为了保持和发展我国在高能物理 τ 粲能区实验研究的领先地位, 我国政府又拨巨资对北京正负电子对撞机和北京谱仪进行升级改进, 称为 BEPCII 和 BESIII. BEPCII 的设计指标是产生粒子反应的强度约为原对撞机的 100 倍, BESIII 的性能则比原北京谱仪有大幅度的提高. 目前, BEPCII 和 BESIII 已经完成安装, 并在 2008 年开始实验取数. 有理由期望, 利用升级改进后的 BESIII, 可以获得比原北京谱仪更多、更精细、更重要的物理成果. 为了达到这一目标, 应用比原北京谱仪数据分析更为精细、更为有效的多元统计分析方法成为一个十分重要和急迫的任务. 事实上, 多元统计分析方法应用于高能物理实验数据分析近年来已经成为国际高能物理界的一种普遍趋势.

本书对于实验数据分析中, 特别是高能物理实验数据分析中涉及的多元统计分析方法作一概略的介绍. 重点讨论统计识别的基本原理以及进行统计识别的具体方法; 对于复杂的数学理论, 只介绍其结果, 而不作深奥的证明. 目的是希望读者能够通过本书掌握多元统计分析的方法并将其付诸实施, 特别是能在 BESIII 的数据

分析中起到一定的作用.

作者诚恳希望得到专家和读者的批评和指正.

朱永生

2008 年 5 月

目　录

第一章　绪　　论

复杂大系统的科学研究取决于对系统结构、性能深刻透彻的认识, 系统研究对象运动规律的掌握, 以及系统运动规律的准确判断和预见. 复杂大系统的科学研究往往都需要收集和处理大量反映系统特征和运行状态的数据信息, 这类原始数据集合由于样本数量巨大, 刻画系统特征的指标变量众多, 并且带有随机性质, 从而形成了规模宏大、复杂难辨的数据海洋. 如何认识和分析高维复杂数据集合中的内在规律性, 简捷地把握系统的本质特征; 如何对高维复杂数据集合进行综合、变换, 将隐藏在其中的重要信息集中提取出来; 如何充分发掘数据中的丰富内涵, 清晰地展示系统的结构特征和系统元素间的内在联系, 直观地描绘系统的运动过程; 这些都是复杂大系统的科学研究取得正确的科学成果的基础和有效工具. 利用统计学和数学方法对多维复杂数据集合进行科学分析的理论和方法, 就是多元统计分析研究的基本内容.

大型高能物理实验就是典型的复杂大系统的科学研究工作. 多元统计分析方法应用于高能物理实验数据分析近年来已经成为一种趋势. 本书对于实验数据分析中, 特别是高能物理实验数据分析中涉及的多元统计分析问题作一概略的介绍. 对于多元统计分析更广泛和深入的了解, 可参考有关的文献和书籍[1~7]. 为了解多元统计分析方法所需的概率和数理统计知识, 可参考文献和书籍[8~11].

1.1　模式和模式识别

我们在生活中时刻都在自觉或不自觉地进行模式识别. 回顾四周, 我们会认出熟识的家人和不认识的陌生人, 能认出周围的物体是椅子还是计算机; 听到声音, 能分辨出是演奏音乐还是汽车在街上奔驰; 闻到气味, 能区分是花的芳香还是炸带鱼的腥味 …… 凡此种种, 都因为人类具备模式识别的能力.

广义地说, 存在于时间和空间中的可观察事物, 它所具有的特定的形态或信息, 都可以称之为模式. 不同的事物可以有截然不同的或者相似的形态特征, 因而可以区别它们是否不同或者是否相似. 通常, 把每个个体具有的特定的形态或信息称为模式, 而具有相似形态的不同个体的集合称为模式类 (或简称为类). 另一种习惯的说法是将模式类称为模式, 而把该模式类中个别的具体模式称为样本. 这种用词的不同可以从上下文分清其含义而不致混淆.

所谓模式识别, 就是将观测到的某一具体事物正确地归入某一类别.

模式类可以有不同的级别. 如自然界的生物物种可以区分为动物、植物和微生物, 动物中有鱼类、鸟类之分等等. 模式识别一般是在同一级别的模式类中将不同样本区分为不同的子类. 例如可以有这样的命题: 怎样区分公羊与母羊, 怎样区分雄性动物与雌性动物; 而不会有这样的命题: 怎样区分公羊与雌鱼.

模式识别在科学研究中, 特别是在实验数据的分析中具有广泛的应用.

对特定的一个或若干个过程进行实验测量, 其目的通常是研究产生这些过程的物理机制, 或者是寻找新的物理现象. 例如在北京正负电子对撞机的北京谱仪实验中, 通过研究正负电子对撞产生的下述反应

$$e^+e^- \to \psi(2S) \to \tau^+\tau^- \to e^\pm\mu^\mp\nu_e\bar{\nu}_e\nu_\mu\bar{\nu}_\mu \tag{1.1.1}$$

来研究 τ 轻子对的产生[12]. 实验给出了 $\psi(2S) \to \tau^+\tau^-$ 衰变分支比的世界首次测量值. 北京谱仪实验中, 正负电子会产生大量的反应过程, 式 (1.1.1) 所示的过程只是其中极小的一部分. 对于该项研究, 式 (1.1.1) 所示的过程是需要寻找的反应模式, 称为信号模式, 或简称为信号, 由该反应模式产生的事例称为信号事例; 大量存在的所有其他的反应模式, 称为本底模式, 或简称为本底. 该过程的数据分析, 实际上就是根据实验数据把实验中产生的所有反应事例分类为信号事例和本底事例的过程和方法, 是一种特定形式的模式识别. 这一类的模式识别的过程和方法在高能物理实验研究乃至一般的科学研究中具有典型意义.

1.2　模式识别系统

有两种基本的模式识别方法: 统计模式识别方法和结构 (句法) 模式识别方法. 高能物理实验的研究对象都是随机过程和随机变量, 它们都服从相应的统计分布, 所以这里只讨论统计模式识别方法. 对于大多数科学实验, 观测量的测量大多存在具有统计性质的误差, 也适用统计模式识别方法.

模式识别系统由两个过程组成, 即设计和实现. 设计是指用一定数量的样本 (训练集或学习集) 进行分类器的设计. 实现是指用设计好的分类器对待识别的样本进行分类决策. 这样的模式识别称为监督模式识别, 即有训练样本情况下的模式识别. 统计模式识别系统主要由四部分组成: 数据获取、预处理、特征选择和分类决策. 如图 1.1 所示.

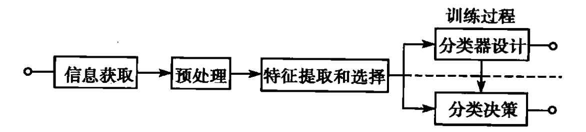


图 1.1　模式识别系统的基本构成

1.2.1 原始数据获取

现代高能物理实验通常利用大型探测装置对研究对象 (如加速器或宇宙线产生的粒子反应) 进行测量, 实验得到的是探测装置对研究对象所记录的大量原始数据, 它们包含了研究对象的模式信息[13]. 如果我们知道了一个反应事例的初态和末态所有粒子的种类、动量和能量, 我们就获得了该反应事例的所有可观测的信息. 因此高能物理实验探测装置的测量目的就在于得到所发生的所有反应事例中粒子的种类、动量和能量. 探测装置能够直接测量的基本粒子必须满足一定的条件: 它们必须是稳定的, 或者有比较长的寿命, 以至于可以在探测装置中飞过比较长的距离; 它们应当与探测装置中的物质有相互作用, 以至于可以被探测装置所测量, 产生测量信号. 这样的基本粒子只有相当有限的几种, 最常见的是

$$\gamma,\quad e^{\pm},\quad \mu^{\pm},\quad \pi^{\pm},\quad K^{\pm},\quad p,\quad \bar{p}. \tag{1.2.1}$$

高能物理实验的直接观测量是探测装置 (及其电子学) 对于每个反应事例中的所有粒子的响应输出信号, 一般分为时间 (TDC) 信息和幅度 (ADC) 信息. 由于一个实验收集的反应事例数量极大, 它们只能用高速计算机在线地记录和存储起来, 以供今后进行离线的物理分析.

1.2.2 原始数据的预处理

高能物理实验探测装置直接观测记录的 TDC 和 ADC 原始数据虽然包含了每个事例的全部可观测信息, 但它们只是这些信息的间接反映, 不能直接地反映粒子反应的 "面貌" 和性质, 不能直接用来作物理分析. 将这些直接记录的 TDC 和 ADC 原始数据转化为能够直接反映粒子反应性质的物理数据的过程称为预处理. 高能物理实验中的原始数据的预处理一般包括刻度和重建, 这当然需要对该实验装置和实验研究目标的透彻了解, 这里不作介绍, 有兴趣的读者可以阅读文献 [13] 及相关的文献.

1. "直接" 实验信息

直接观测量通过预处理后, 一般转化为: 带电径迹的空间飞行轨迹和飞行时间 (time-of-flight, 即 TOF) 信息, 带电径迹的空间飞行轨迹结合磁场的数据可以得到带电径迹的动量; 带电粒子电离能损的信息, 它和 TOF 信息都可以用来作带电粒子种类的鉴别; 具有电磁和强子量能器的探测装置可以给出电磁 $(\gamma, e^{\pm})$ 粒子、$\mu^{\pm}$ 和强子 $(\pi^{\pm}, K^{\pm}, p, \bar{p})$ 的簇射沉积能量和簇射形态的信息.

2. "间接" 实验信息

利用这些 "直接" 实验信息, 还可以推导得到 "间接" 实验信息.

(1) 事例的初级顶点

一个事例如果产生 2 条以上的带电径迹, 由这些带电径迹的交点可求得事例的初级顶点, 在正负电子对撞实验中, 初级顶点相应于正负电子对撞点的位置.

(2) 短寿命粒子存在的信息

一些粒子的寿命极短, 一旦产生几乎立即衰变为两个或更多的粒子, 典型的例子如 $\pi^0 \to \gamma\gamma, \eta \to \gamma\gamma, \omega \to \pi^+\pi^-\pi^0$ 等. 短寿命粒子存在的信息可由所谓的不变质量得到. 粒子物理告诉我们, 若粒子 A(质量 M) 衰变为 j 个粒子

$$\mathrm{A} \to 1 + 2 + \cdots + j. \tag{1.2.2}$$

各粒子的四动量分别记为 $p_A, p_1, \cdots, p_j$. 粒子四动量定义为一个四维矢量 $p=(E, \mathrm{i}\boldsymbol{p})$, E 为粒子能量, $\boldsymbol{p}$ 为粒子的动量. 这 j 个粒子的四动量之和的平方称为它们的不变质量 (或有效质量) 平方, 并恰好等于母粒子 A 的质量平方:

$$M^2 \equiv \left(\sum_j p_j\right)^2 = \left(\sum_j E_j\right)^2 - \left(\sum_j \boldsymbol{p}_j\right)^2. \tag{1.2.3}$$

它是洛伦兹变换下的不变量, 即在不同的惯性系中 M^2 值不变. 按照这一性质, 可以根据两个光子的不变质量是否等于 π^0 或 η 的质量来判断 π^0 或 η 是否存在, 根据 $\pi^+\pi^-\pi^0$ 的不变质量是否等于 ω 的质量来判断 ω 是否存在, 等等.

(3) 长寿命粒子存在的信息, 次级顶点

一些粒子的寿命比较长, 它们产生以后要飞行一段距离之后才衰变成两个或更多的粒子. 这类粒子存在的信息可由它们衰变的次级顶点给出. 不稳定粒子衰变时间为 t 的概率密度为

$$f(t) = \frac{1}{\tau}\mathrm{e}^{-t/\tau},$$

式中, τ 是不稳定粒子的平均寿命. 相应于衰变时间 t, 粒子的飞行距离 $l = t\gamma\beta c$. 典型的例子如 $\mathrm{K}_S^0 \to \pi^+\pi^-(c\tau = 2.6842\mathrm{cm})$, $\Lambda \to \mathrm{p}\pi^-(c\tau = 7.89\mathrm{cm})$, 它们在北京谱仪实验中的典型飞行距离为厘米量级. 这样 Λ 衰变产生的 p, π^- 两根径迹的交点离正负电子对撞中心 (初级顶点) 有一定的距离, 被称为次级顶点. 如果收集大量的动量相同的 $\Lambda \to \mathrm{p}\pi^-$ 事例, 次级顶点到初级顶点间的距离应当服从指数分布. 对于 $\mathrm{K}_S^0 \to \pi^+\pi^-$ 衰变, 情形是类似的. 因此, 在研究末态包含长寿命粒子的反应时, 次级顶点位置也常常作为粒子反应的一个重要输入量.

(4) 不可探测粒子存在的信息

一些粒子与探测器物质 (几乎) 不发生作用, 这样探测器不能给出它们存在的直接信号. 在北京谱仪正负电子对撞实验中, 属于这类粒子有 $\nu, \mathrm{K}_L^0, \mathrm{n}, \bar{\mathrm{n}}$ 等等. 它们的存在信息可用丢失质量或丢失能量给出.

若粒子 A(已知质量为 M) 衰变为 3 个粒子

$$\mathrm{A} \to 1 + 2 + 3 \tag{1.2.4}$$

其中粒子 1,2 是可探测粒子, 测量到的能量和动量为 E_i 和 $\boldsymbol{p}_i, i=1,2$. 粒子 3 是不可探测粒子, 那么粒子 3 的质量 (如果粒子 3 是 0 质量粒子, 如中微子, 则为粒子 3 的能量) 等于

$$M_3 = M_{12}^{\mathrm{mis}} = \left[(M - E_1 - E_2)^2 - (\boldsymbol{p}_1 + \boldsymbol{p}_2)^2\right]^{1/2}. \tag{1.2.5}$$

如北京谱仪实验中, 粒子反应 $\psi(2\mathrm{S}) \to \mathrm{p}\pi^-\bar{\mathrm{n}}$ 的不可探测粒子 $\bar{\mathrm{n}}$ 的存在可利用可探测粒子 p, π^- 的丢失质量是否与 $\bar{\mathrm{n}}$ 的质量相接近来推断. 因此, 在研究末态包含不可探测粒子的反应时, 丢失质量往往是输入变量之一.

3. *反应事例的实验数据*

一般说来, 对于一个记录到的反应事例, 它的末态粒子的以下实验信息构成该事例的实验数据:

带电径迹的数目;

每根带电径迹的 TOF 和 $\mathrm{d}E/\mathrm{d}x$ 信息;

每根带电径迹的动量;

γ 光子的数目;

所有可探测粒子的簇射沉积能量和簇射形态的信息;

初级顶点位置;

次级顶点位置 (如需要);

不变质量 (如需要);

丢失质 (能) 量 (如需要);

……

一个实验收集的所有反应事例的实验数据构成该实验的实验数据集.

一个事例所记录的全部实验数据 (假定是 n_r 个) 可以看成是一个 n_r 维向量, 每一个分量是该事例的一个有效物理量的表征. 由于粒子反应都是随机过程, 每一个这样的物理量都是随机变量, 具有各自的概率分布. 每一个事例的这个 n_r 维向量的具体数值是 n_r 维随机向量的一个实现, 或者说一个样本, 它可以用测量空间 (n_r 维) 中的一个点来表示, 于是实验数据集转化为测量空间中的一个数据点集, 它是实验测量数据 n_r 维随机向量总体分布的一个实现, 是进行进一步物理分析的基础. 高能物理实验的数据向量的维数 n_r 往往达到几十或者上百, 一个实验收集的反应事例数往往达到 $10^6 \sim 10^{10}$ 量级.

1.2.3 特征提取和选择

由于高能物理实验的测量空间中的数据点集数量庞大, 为了有效地进行分类识别, 就要对实验数据进行筛选和变换, 得到最能反映分类本质的特征物理量, 这就是特征提取和选择的过程. 特征提取和选择后确定的物理量构成的空间称为特征空间, 于是测量空间中的数据向量转化为特征空间中的数据向量, 测量空间中的数据点集转化为特征空间中的数据点集, 它成为进行事例分类的直接输入变量. 在本书以后的陈述中, 除非特别说明, 作为事例分类器直接输入变量的特征向量也称为数据向量.

特征提取和选择应遵循三个原则, 第一是有效性, 即提取的物理量应该能够有效地区分信号和本底; 第二是充分性, 即提取的物理量能够完整地保留事例的全部有用信息; 第三是具有降维能力, 即通过变换, 可把维数较高的测量空间 (n_r 维) 中的模式变为维数较低的特征空间 (n 维) 中的模式, 这就能有效地减少分类器设计和应用它作分类决策所需的计算量. 特征空间中的数据向量是由测量空间中的数据随机向量通过变换得到的, 因而它也是随机向量.

应当指出, 大型的科学实验一般具有多重研究对象和科学目标, 因此实验数据向量的维数 n_r 需要足够高, 以能包含充分多的实验信息供各种研究课题的需要. 但对于某一特定课题而言, 只需提取和选择与该课题有关的 n 维变量作为特征变量就可以作出正确的分类, 一般 $n < n_r$. 例如, 为了区分人的性别, 只需要考察人类性体征特点就可以了, 没有必要对与此无关的其他体征进行比较分类. 同样, 对于同一个高能物理实验中不同反应过程的分析, 只要选择与各自过程相关的物理量作为各自的特征变量, 这样就大大降低特征空间的维数, 从而大大降低分析的困难程度, 节省计算的时间. 这对于具有庞大数量事例数的高能物理实验极为重要.

1.2.4 分类决策

分类决策就是在特征空间中用统计方法把被识别的对象归为某一类别, 基本做法是根据样本训练集的特征变量的行为确定某个或若干个判据, 使得按照这种判据对识别对象进行分类得到的效率 (正确分类的比例) 最高, 误判率最低.

由于实验特征空间中的数据向量是多维随机向量, 这就决定了基于这类数据的分类决策过程是多变元统计分析的过程.

高能物理实验中的模式识别, 就是将观测到的每一个事例正确地归入某一粒子反应类别. 但是, 对于一项具体的研究而言, 研究者感兴趣的粒子反应可能只有一种或几种. 一般称为信号, 此外的反应过程都称为本底.

1. *粒子鉴别和事例判选*

高能物理实验的数据分析的目的是把信号事例从大量本底事例中挑选出来 (称

为事例判选), 然后对信号事例的性质进行进一步的研究. 事例判选一般经过两个步骤: ① 粒子鉴别, ② 反应过程拓扑形态的鉴别. 以式 (1.1.1) 所示的反应为例, 首先我们要确定末态粒子是 1 个电子和 1 个 μ 子 (实验一般不直接测量中微子), 由于实验可观测的粒子种类有式 (1.2.1) 所列的几种, 所以首先要从中确定所研究末态的粒子种类 (本例中是电子和 μ 子), 这是一个多总体的模式识别问题, 或者说是多类模式的判别问题. 其次, 我们要确定反应确实是通过中间态 $\tau^+\tau^-$ 再到达 $e^\pm\mu^\mp\nu_e\bar{\nu}_e\nu_\mu\bar{\nu}_\mu$ 末态, 这就是一个反应过程拓扑形态的识别问题, 识别的结果总是将所有的事例区分为信号事例和本底事例, 是一个 2 个总体的模式识别问题, 或者说是两类模式的判别问题. 尽管有时粒子鉴别和反应拓扑形态的鉴别不一定截然分明, 但是这两类判别问题在事例判选中总是存在的. 一个好的粒子鉴别判据 (粒子分类器) 应当对粒子有高的正确判定效率, 有低的误判率. 一个好的事例判选判据 (事例分类器) 应当对信号事例有高的选择效率, 有低的误判率 (即对本底事例有低的选择效率或高的排除率).

2. *样本训练*

对于一个测量到的粒子信息, 怎样判定它是式 (1.2.1) 中的哪一种粒子呢? 解决这个粒子鉴别问题需要采用对已知样本进行训练的方法. 具体地说, 就是利用已知是 $e^\pm$ 粒子的数据样本 $\boldsymbol{X}_{e,N_e\times n}$(下标中的 e 表示电子, N_e 表示电子样本的个数, n 表示用 n 个特征变量表征该电子样本. 见 1.3.1 节关于数据矩阵的定义), 已知是 $\mu^\pm$ 粒子的数据样本 $\boldsymbol{X}_{\mu,N_\mu\times n}$, 以及已知是 $\gamma,\pi^\pm,K^\pm,p,\bar{p}$ 的数据样本 $\boldsymbol{X}_{\gamma,N_\gamma\times n},\cdots$, 根据这几类数据样本的差异寻找出一组判据, 使得它对每种粒子都有高的正确判定效率, 有低的误判率. 寻找这组判据的过程称为训练 (或学习) 过程, 实际上就是分类器的设计过程. 然后, 对于一个测量到的粒子 (种类待定) 信息, 应用该判据来判定它是何种粒子. 这也就是用设计好的分类器对待识别的样本进行分类决策.

类似地, 对于一个测量到的事例信息, 要判断它是不是某个特定的信号事例, 需要利用已知是该信号事例的数据样本和已知是它的本底事例的数据样本进行训练, 根据这两类数据样本的差异寻找出一组判据, 使得它对信号事例有高的正确判定效率, 有低的误判率. 然后, 对于一个测量到的事例信息, 应用该判据来判定它是信号事例或本底事例.

3. *训练样本的获得*

我们看到, 高能物理实验数据分析中的粒子鉴别和事例判选的实现, 首先要有各种粒子的数据样本和各种粒子反应事例的数据样本. 这两类数据样本有两种途径可以得到: 蒙特卡罗模拟数据和真实实验数据.

a. 蒙特卡罗模拟数据

先讨论粒子反应的蒙特卡罗模拟数据. 假定我们要研究的是 bhabha 事例, 即

$e^+e^- \to e^+e^-$ 反应事例, 所谓粒子反应的蒙特卡罗模拟数据, 首先是根据粒子物理理论的预期和反应初态正负电子的四动量 (已知值), 计算出末态正负电子的四动量. 这个过程由反应的产生子来完成, 它依赖于粒子物理对所研究的反应的理论了解. 粒子物理界对于不少粒子反应已经有相当透彻的了解, 有了相应的事例产生子可以使用, 特别是对电磁相互作用过程有很精确的理论描述, 因此电磁相互作用过程的产生子一般比较精确可信. 比较起来, 粒子物理对于强作用的理论描述要粗糙得多, 因此涉及强作用的粒子反应的产生子的精确性比较差.

知道了反应末态粒子的四动量, 让末态粒子按照自己的动量和方向进入探测器, 与探测器中的物质发生作用. 粒子与物质的相互作用也是按照粒子物理的各种理论模型来描述的, 这一过程十分复杂. 目前粒子物理学界通用的是 Geant 程序框架[14], 它汇集了人类对于粒子与物质的各种相互作用至今所了解的知识. 这种相互作用的结果, 就得到了探测器对于该反应末态粒子的探测信号. 这一切都是通过计算机利用理论所提供的模拟数学公式进行计算得到的, 所以称为模拟计算, 得到的数据称为粒子反应的蒙特卡罗模拟数据. 这种计算的过程好像是用计算机作物理实验. 这个过程在高能物理实验数据分析中称为探测器模拟. 如果理论所提供的数学公式是正确的, 那么所得到的粒子反应的蒙特卡罗模拟数据与粒子反应的真实实验数据应当是接近的.

为了把信号事例从实验中产生的全部事例中挑选出来, 不但要有信号事例的产生子, 还需要有实验中产生的所有反应的事例产生子. 对于正负电子对撞实验而言, 就是要有 $e^+e^- \to$ 所有可产生过程的事例产生子. 所谓的 LUND 模型提供了这样的产生子[15]. 对于其他的粒子反应研究, 亦需要相应的所有可产生过程的事例产生子. 一般这类产生子所依据的理论模型比较粗糙, 与实验中的真实情况有所差别. 所以基于这种产生子确定的信号/本底事例判别条件以及相应的信号/本底事例误判率与实际的信号/本底事例误判率存在差异, 在实验数据分析中, 必须考虑这种差异导致的系统误差.

各种粒子的数据样本的获得则比较简单, 任意粒子的产生器都是十分容易构造的, 再通过探测器模拟就得到该粒子的蒙特卡罗模拟数据.

蒙特卡罗模拟数据样本的好处是样本量可以任意地大 (只要计算机能力允许). 它的缺点是数据样本的正确性和精确性受到理论模型的正确性和精确程度的限制, 同时它不能反映探测器电子学噪声和束流管道中正负电子束流–气体相互作用本底带来的对真实数据的影响, 即使加入了这种噪声和束流–气体相互作用本底的模拟, 由于模拟公式往往缺乏理论根据或者十分粗糙, 也不一定能反映真实情况.

b. 真实实验数据

所谓粒子反应的真实实验数据, 就是通过一定的事例判选把某种粒子反应事例判选出来. 例如可以通过某些判据把辐射 bhabha 事例, 即 $e^+e^- \to \gamma e^+e^-$ 反应事

例判选出来. 这种事例的末态电子和 γ 光子可以具有 $(0 \sim E_{\rm b})$ 各种能量 ($E_{\rm b}$ 是初态电子束流能量), 且具有各种方向. 这样, 我们就获得了各种能量、各种方向的电子和 γ 光子的真实实验数据, 可以作为粒子鉴别的训练样本. 又比如可以通过某些判据把 $e^+e^- \to J/\psi \to \rho\pi \to \pi^+\pi^-\pi^0 \to \pi^+\pi^-\gamma\gamma$ 事例判选出来, 末态的两个带电粒子是具有各种方向、各种能量的 π^+,π^- 介子, 这样, 我们就获得了各种能量、各种方向的 π^+,π^- 介子的真实实验数据, 可以作为粒子鉴别的训练样本. 类似地, 我们可以通过适当判据把末态包含式 (1.2.1) 所列粒子的粒子反应事例判选出来, 获得这些粒子的真实实验数据, 作为粒子鉴别的训练样本.

真实实验数据的数量受到实验收集的总事例数和粒子反应截面的限制. 如果反应截面很小, 相应的反应事例只占收集的总事例数的很小一部分, 实验收集的总事例数又不够大, 那么反应末态粒子的数量就不大. 对于统计分析而言, 就可能造成较大的统计涨落. 另一方面, 通过某些判据把一种特定的反应事例判选出来, 可能存在误判, 即混有其他本底事例, 样本不纯. 为了避免这种本底污染, 往往把事例判选判据设定得严一些, 降低误判率, 这样作的结果提高了样本的纯度, 但牺牲了统计量.

高能物理实验数据分析中, 作粒子鉴别时的训练样本应该尽可能使用真实实验数据样本, 而作事例判选时信号事例的训练样本一般是蒙特卡罗模拟数据样本, 因为在完成信号事例的判选之前, 不可能获得信号事例的真实实验数据样本. 用实验收集的全部事例的真实实验数据样本, 扣除可能的信号事例样本 (利用蒙特卡罗模拟数据样本确定的信号事例判选条件来选择) 后, 可作为本底事例的训练样本.

1.3 数据矩阵与样本空间

1.3.1 数据矩阵与样本空间

前面已经提到, 事例分类的直接输入变量是 n 维特征空间中的数据向量, 每一个 n 维数据向量包含了一个特定事例的所有可观测的信息, 或者说代表了一个特定的事例. 假定我们有 N 个事例, 要将它们区分为信号事例和本底事例. 这 N 个事例构成 n 维特征空间中的 N 个样本点. 于是输入数据可表示为如表 1.1 所示的形式

表 1.1 输入数据表

样本 \ 特征变量	$\boldsymbol{e}_1$	$\boldsymbol{e}_2$	$\cdots$	$\boldsymbol{e}_j$	$\cdots$	$\boldsymbol{e}_n$
$\boldsymbol{x}_1$	x_{11}	x_{12}	$\cdots$	x_{1j}	$\cdots$	x_{1n}
$\boldsymbol{x}_2$	x_{21}	x_{22}	$\cdots$	x_{2j}	$\cdots$	x_{2n}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\boldsymbol{x}_i$	x_{i1}	x_{i2}	$\cdots$	x_{ij}	$\cdots$	x_{in}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\boldsymbol{x}_N$	x_{N1}	x_{N2}	$\cdots$	x_{Nj}	$\cdots$	x_{Nn}

或表示为矩阵形式

$$\boldsymbol{X}_{N\times n}=\begin{pmatrix} x_{11} & x_{12}\cdots x_{1n} \\ x_{21} & x_{22}\cdots x_{2n} \\ \vdots & \vdots & \vdots \\ x_{N1} & x_{N2}\cdots x_{Nn} \end{pmatrix}_{N\times n}=\begin{pmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \boldsymbol{x}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_N^{\mathrm{T}} \end{pmatrix}=(\boldsymbol{e}_1,\boldsymbol{e}_2,\cdots,\boldsymbol{e}_n) \tag{1.3.1}$$

式中

$$\boldsymbol{x}_i=(x_{i1},x_{i2},\cdots,x_{in})^{\mathrm{T}},\qquad i=1,2,\cdots,N, \tag{1.3.2}$$

表示 N 个样本点, 这 N 个样本点组成了一个点群集合. 所有的样本点所占据的空间构成了 (n 维) 样本空间或特征空间 $\boldsymbol{F}\in\boldsymbol{R}^n$. 每个样本点向量 $\boldsymbol{x}_i$ 称为特征向量, 它的 n 个分量表示事例 i 的 n 个特征物理量, 例如事例的带电径迹数, 带电径迹的动量等等.

数据矩阵的每一列描述一个变量 $\boldsymbol{e}_j$,

$$\boldsymbol{e}_j=(x_{1j},x_{2j},\cdots,x_{Nj})^{\mathrm{T}},\qquad j=1,2,\cdots,n. \tag{1.3.3}$$

它表示 N 个事例的第 j 个特征物理量的测量数值. 它是一个随机变量, 因此有其统计特征, 如均值 (或期望值)、方差、协方差、相关系数等. 所有变量的集合构成 (N 维) 变量空间 $\boldsymbol{E}\in\boldsymbol{R}^N$. 可以用样本统计量来估计随机变量的数字特征.

变量 $\boldsymbol{e}_j$(第 j 个特征物理量) 的均值 $\bar{x}_j$

$$\bar{x}_j=\frac{1}{N}\sum_{i=1}^{N}x_{ij}, \tag{1.3.4}$$

方差 s_j^2

$$s_j^2=\frac{1}{N-1}\sum_{i=1}^{N}(x_{ij}-\bar{x}_j)^2, \tag{1.3.5}$$

变量 $\boldsymbol{e}_j$ 与变量 $\boldsymbol{e}_k$ 的协方差 s_{jk}

$$s_{jk}=\frac{1}{N-1}\sum_{i=1}^{N}(x_{ij}-\bar{x}_j)(x_{ik}-\bar{x}_k) \tag{1.3.6}$$

它用于测度变量 $\boldsymbol{e}_j$ 与 $\boldsymbol{e}_k$ 的相关性. 写成矩阵形式

$$V=\begin{pmatrix} s_{11} & s_{12} & \cdots s_{1n} \\ s_{21} & s_{22} & \cdots s_{2n} \\ \vdots & \vdots & \vdots \\ s_{n1} & s_{n2} & \cdots s_{nn} \end{pmatrix}, \tag{1.3.7}$$

称为样本的协方差矩阵. 相关系数 r_{jk}

$$r_{jk}=\frac{s_{jk}}{s_j s_k},\tag{1.3.8}$$

它满足 $-1\leqslant r_{jk}\leqslant 1$, r_{jk} 量纲为一, 可更准确地表征两个变量间的相关性.

1.3.2 模式的相似性度量

尽管不同的模式识别理论与方法之间存在差异, 但模式的所有分类与描述都是以若干公认的假设 (公设) 为基础的. 其中关于模式的相似性公设可陈述为：如果两个模式的特征或其简单的组成部分仅有微小差别, 则称两个模式相似.

最简单和直观的分类方法, 是直接以不同类的训练样本点的集合所构成的区域表示各类决策区, 并以角距离或点距离作为样本点相似性 (类似度) 度量的主要依据.

1. *样本点矢量间夹角余弦*

当不同模式类的样本点呈扇状分布时 (见图 1.2), 可用夹角余弦定义两样本点的角距离, 即角度相似性函数：

$$s(\boldsymbol{x},\boldsymbol{y})=\cos\theta=\frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}}{|\boldsymbol{x}||\boldsymbol{y}|},\tag{1.3.9}$$

其中, θ 为两样本点矢量 $\boldsymbol{x},\boldsymbol{y}$ 之夹角; $|\boldsymbol{x}|$ 为矢量 $\boldsymbol{x}$ 的模. 夹角越接近于 0 (夹角余弦越接近于 1), 两样本点越相似. 即若 $s(\boldsymbol{x},\boldsymbol{y})>s(\boldsymbol{x},\boldsymbol{z})$, 则认为 $\boldsymbol{x}$ 与 $\boldsymbol{y}$ 更相似些.

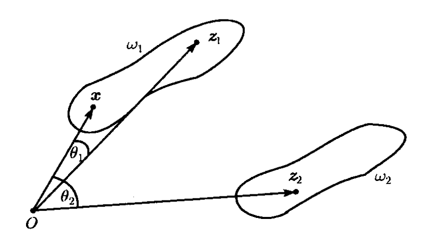


图 1.2 用夹角余弦定义两样本点的相似性

2. *样本点间的距离*

样本点间的距离常常作为样本间相似性的一种度量, 即两个样本点间的距离越近, 这两个样本越相似. 一般, 所选的距离函数应满足下列条件：

$$\begin{aligned}&d(\boldsymbol{x},\boldsymbol{y})=d(\boldsymbol{y},\boldsymbol{x}),\\&d(\boldsymbol{x},\boldsymbol{y})\leqslant d(\boldsymbol{x},\boldsymbol{z})+d(\boldsymbol{y},\boldsymbol{z}),\\&d(\boldsymbol{x},\boldsymbol{y})\geqslant 0,\\&d(\boldsymbol{x},\boldsymbol{y})=0,\qquad \text{当且仅当 } \boldsymbol{x}=\boldsymbol{y} \text{ 时}.\end{aligned}\tag{1.3.10}$$

根据不同的应用目的, 已提出多种满足以上条件的距离函数, 这里仅列出常用的几种:

(1) Minkowsky 距离

$$d(\boldsymbol{x},\boldsymbol{y})=\left[\sum_{j=1}^{n}|x_j-y_j|^{\lambda}\right]^{1/\lambda}. \tag{1.3.11}$$

(2) Manhattan 距离

$$d(\boldsymbol{x},\boldsymbol{y})=\sum_{j=1}^{n}|x_j-y_j|, \tag{1.3.12}$$

这是 Minkowsky 距离 $\lambda=1$ 时的特例.

(3) Euclidean (欧氏) 距离

$$d(\boldsymbol{x},\boldsymbol{y})=\left[\sum_{j=1}^{n}|x_j-y_j|^{2}\right]^{1/2}, \tag{1.3.13}$$

这是 Minkowsky 距离 $\lambda=2$ 时的特例.

上述距离使用时要注意样本各输入变量分量的量纲. 例如某一样本的两个输入分量分别为长度和压力, 若将长度单位由毫米改成厘米, 压力单位由厘米汞柱改成毫米汞柱, 则分类时压力的影响较改变前将大为增加.

(4) Mahalanobis (马氏) 距离

$$d^2(\boldsymbol{x},\boldsymbol{\mu})=(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{V}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}), \tag{1.3.14}$$

其中, $\boldsymbol{\mu}$ 为总体的均值向量; $\boldsymbol{V}$ 为相应的协方差矩阵.

马氏距离考虑了样本的各输入变量分量的统计特性, 特别是考虑了各输入变量分量的相关性影响; 而上列的其他距离均没有考虑各输入变量分量的相关性. 当协方差矩阵 $\boldsymbol{V}$ 为对角矩阵时, 各分量相互独立; 特别当协方差矩阵 $\boldsymbol{V}$ 为单位矩阵时, 马氏距离与欧氏距离相等.

以上的各种距离度量在实际应用中, 在计算的复杂性方面, 在是否便于进行解析分析方面效果各不相同. 由于欧氏距离在许多情况下便于分析和计算, 因此常常被各种分类器采用.

1.3.3 样本点的权重和特征向量数据的预处理

1. 样本点的权重

一般情况下, 如果认为每个样本点的重要性是相等的, 则对每一个样本点赋予同样的权重 $w_i=1/N, i=1,2,\cdots,N$. 但是, 如果每个样本点的抽取是不等概率的,

那么, 每一个样本点的权重 w_i 可以是不同的. 例如, 在进行民意测验时, 人口较多的地区的调查数据 (样本点) 应当比人口较少的地区的调查数据 (样本点) 有更大的权重. 所有样本点的权重之和 (总权重) 应当等于 1, 即

$$\sum_{i=1}^{N} w_i = 1. \tag{1.3.15}$$

考虑样本点的权重后, 相应的均值、方差、协方差的定义需改写为:
变量 $\boldsymbol{e}_j$ 的 (加权) 平均 $\bar{x}_j$

$$\bar{x}_j = \sum_{i=1}^{N} w_i x_{ij}, \tag{1.3.16}$$

方差 s_j^2

$$s_j^2 = \sum_{i=1}^{N} w_i (x_{ij} - \bar{x}_j)^2, \tag{1.3.17}$$

变量 $\boldsymbol{e}_j$ 与变量 $\boldsymbol{e}_k$ 的协方差 s_{jk}

$$s_{jk} = \sum_{i=1}^{N} w_i (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k). \tag{1.3.18}$$

2. 特征向量数据的中心化

特征向量数据的中心化是对数据作平移变换:

$$x_{ij}^* = x_{ij} - \bar{x}_j, \qquad i = 1, 2, \cdots, N, \qquad j = 1, 2, \cdots, n. \tag{1.3.19}$$

该变换使新坐标系的原点 O^* 与数据点群的重心重合, 而不改变样本点间的相互位置, 也不改变数据变量各分量间的相关性, 但带来计算上的许多便利.

定义变量空间 $\boldsymbol{E}$ 中的度量矩阵 $\boldsymbol{D}$ 为

$$\boldsymbol{D} = \begin{pmatrix} w_1 & 0 & \cdots 0 \\ 0 & w_2 & \cdots \\ \vdots & & \vdots \\ 0 & \cdots & w_N \end{pmatrix} \equiv \mathrm{diag}(w_1, w_2, \cdots, w_N). \tag{1.3.20}$$

对于变量空间中的任意两个矢量 $\boldsymbol{e}_j = (x_{1j}, x_{2j}, \cdots, x_{Nj})^{\mathrm{T}}$ 和 $\boldsymbol{e}_k = (x_{1k}, x_{2k}, \cdots, x_{Nk})^{\mathrm{T}}$, 定义 $\boldsymbol{e}_j$ 和 $\boldsymbol{e}_k$ 的点积为

$$(\boldsymbol{e}_j, \boldsymbol{e}_k)_D = \boldsymbol{e}_j^{\mathrm{T}} \boldsymbol{D} \boldsymbol{e}_k = \sum_{i=1}^{N} w_i x_{ij} x_{ik}. \tag{1.3.21}$$

若考虑样本点的权重，且数据被中心化，$\boldsymbol{e}_j$ 变换为 $\boldsymbol{e}_j^*$，则以下结论成立：

(1) 任意一个中心化处理后的变量 $\boldsymbol{e}_j^*$ 的模等于 $\boldsymbol{e}_j$ 的标准差 s_j：

记中心化处理后的新变量为

$$\boldsymbol{e}_j^* = (x_{1j}-\bar{x}_j, x_{2j}-\bar{x}_j, \cdots, x_{Nj}-\bar{x}_j)^{\mathrm{T}}, \qquad j=1,2,\cdots,n. \tag{1.3.22}$$

则有

$$\|\boldsymbol{e}_j^*\|_D^2 = (\boldsymbol{e}_j^*)^{\mathrm{T}}\boldsymbol{D}\boldsymbol{e}_j^* = \sum_{i=1}^{N} w_i(x_{ij}-\bar{x}_j)^2 = s_j^2, \qquad j=1,2,\cdots,n. \tag{1.3.23}$$

(2) 两个变量 $\boldsymbol{e}_j^*$ 和 $\boldsymbol{e}_k^*$ 的点积等于 $\boldsymbol{e}_j$ 和 $\boldsymbol{e}_k$ 的协方差 s_{jk}：

$$(\boldsymbol{e}_j^*, \boldsymbol{e}_k^*)_D = (\boldsymbol{e}_j^*)^{\mathrm{T}}\boldsymbol{D}\boldsymbol{e}_k^* = \sum_{i=1}^{N} w_i(x_{ij}-\bar{x}_j)(x_{ij}-\bar{x}_k) = s_{jk}. \tag{1.3.24}$$

(3) 两个变量 $\boldsymbol{e}_j^*$ 和 $\boldsymbol{e}_k^*$ 夹角的余弦等于 $\boldsymbol{e}_j$ 和 $\boldsymbol{e}_k$ 的相关系数 r_{jk}：

记 $\boldsymbol{e}_j^*$ 和 $\boldsymbol{e}_k^*$ 的夹角为 θ_{jk}，则

$$\cos\theta_{jk} = \frac{(\boldsymbol{e}_j^*, \boldsymbol{e}_k^*)_D}{\|\boldsymbol{e}_j^*\|_D \cdot \|\boldsymbol{e}_k^*\|_D} = \frac{s_{jk}}{s_j s_k} = r_{jk}. \tag{1.3.25}$$

可见，相关系数的几何含义是数据变量两个矢量在变量空间 $\boldsymbol{E}$ 中的夹角余弦.

(4) 样本点 $\boldsymbol{x}_l^* = (x_{l1}^*, x_{l2}^*, \cdots, x_{ln}^*)^{\mathrm{T}} \in \boldsymbol{R}^n$ 与 $\boldsymbol{x}_m^*$ 之间的欧氏距离与中心化前没有变化：

$$d^2(\boldsymbol{x}_l^*, \boldsymbol{x}_m^*) = \sum_{j=1}^{n}(x_{lj}-x_{mj})^2 = d^2(\boldsymbol{x}_l, \boldsymbol{x}_m), \qquad l,m=1,2,\cdots,N, \tag{1.3.26}$$

3. **特征向量数据的量纲为一化**

由 1.3.3 小节知道，在使用 Minkowski 距离 (包括 Manhattan 距离和欧氏距离) 时，要注意样本输入变量各分量的量纲. 在统计问题中，变量各分量的测度单位往往是不一样的. 例如，当用年工资、年龄和家庭人口来表示输入变量的 3 个分量时，则每个分量的单位完全不同，年工资用百元和用万元作为单位，同样的年工资对于欧氏距离的作用有很大差异. 又比如某个分类问题需要用到身高和头颅长度的数据，如果对它们采用同样的长度单位，身高的变异比较大，对于欧氏距离的贡献就比较大. 实际上身高的变异比较大只是这个变量本身离散程度比较大的反映，简单地用身高的数值来确定欧氏距离而不考虑它的离散性质，实际上夸大了它的作用. 这种由于变量各分量离散程度的不同导致的变异并不反映数据本身的变化情况. 为了消除这种虚假的变异导致的不良影响，就要消除特征向量各分量的量纲效

应, 使每一个分量对样本点间的距离有同等的影响力, 这通常由特征向量数据的量纲为一化处理来达到.

量纲为一化是对数据作变换:

$$x_{ij}^* = x_{ij}/s_j, \qquad i=1,2,\cdots,N, \qquad j=1,2,\cdots,n. \tag{1.3.27}$$

作此变换后得到的新变量为

$$\boldsymbol{e}_j^* = (x_{1j}/s_j, x_{2j}/s_j, \cdots, x_{Nj}/s_j)^{\mathrm{T}}, \qquad j=1,2,\cdots,n. \tag{1.3.28}$$

每个 $\boldsymbol{e}_j^*$ 的方差均等于 1 且量纲为一, 这时, 样本点 $\boldsymbol{x}_i^* = (x_{i1}^*, x_{i2}^*, \cdots, x_{in}^*)^{\mathrm{T}} \in \boldsymbol{R}^n$ 与 $\boldsymbol{x}_m^*$ 之间的欧氏距离平方为

$$d^2(\boldsymbol{x}_l^*, \boldsymbol{x}_m^*) = \sum_{j=1}^{n} \frac{(x_{lj}-x_{mj})^2}{s_j^2} = (\boldsymbol{x}_l - \boldsymbol{x}_m)^{\mathrm{T}} \boldsymbol{M} (\boldsymbol{x}_l - \boldsymbol{x}_m), \qquad l,m=1,2,\cdots,N, \tag{1.3.29}$$

式中, $\boldsymbol{M} = \mathrm{diag}(1/s_1^2, \cdots, 1/s_n^2)$ 是 n 维样本空间的度量矩阵.

4. 特征向量数据的标准化

特征向量数据的标准化是对数据同时作中心化和量纲为一化处理:

$$x_{ij}^* = \frac{x_{ij} - \bar{x}_j}{s_j}, \qquad i=1,2,\cdots,N, \qquad j=1,2,\cdots,n. \tag{1.3.30}$$

作此变换后得到的新变量为

$$\boldsymbol{e}_j^* = \left(\frac{x_{1j}-\bar{x}_j}{s_j}, \frac{x_{2j}-\bar{x}_j}{s_j}, \cdots, \frac{x_{Nj}-\bar{x}_j}{s_j}\right)^{\mathrm{T}}, \qquad j=1,2,\cdots,n. \tag{1.3.31}$$

每个 $\boldsymbol{e}_j^*$ 的均值为 0, 方差等于 1 且量纲为一, 这时, 样本点 $\boldsymbol{x}_l^*$ 与 $\boldsymbol{x}_m^*$ 之间的欧氏距离平方仍由式 (1.3.29) 表示.

应当指出, 是否需要考虑样本集各样本点有不同的权重, 是否需要对特征向量数据作预处理以及作怎样的预处理 (中心化、量纲为一化或标准化), 取决于具体分类问题的要求和所用的分类方法. 如第四章中讨论的决策树林法就需要考虑样本集中各样本点有不同的权重.

1.4 主成分分析

模式识别的分类问题是根据待识别样本的 n 维特征向量的观测值将样本判别为某个类别. 特征向量的每一维变量都是随机变量, 它表征了样本集总体的一个特征. 显然这些特征的选择是很重要的, 它在很大程度上决定了分类器的设计及性能.

假如不同类别的样本集中这些特征的差别很大, 那就比较容易设计出性能较好的分类器.

特征向量的每一维变量往往是直接观测值, 或观测值的组合, 或某种变换得到的物理量. 如在 1.2.2 小节中我们已经列举了粒子物理实验数据分析中的实验数据的组成. 这种实验数据的维数 n_r 往往可大到几十或上百. 对于特定的分类问题, 可根据需要选择其中的部分观测值 (或观测值的组合或某种变换) 作为特征向量的 n 个变量. 即便如此, 人们也往往倾向于取比较大的 n 值, 这与人们的心理因素有关, 总认为特征量越多, 越能包含尽可能多的信息, 便于不同类样本的区分. 但是过多的特征对于一定的模式识别任务来说可能包含许多无用的信息, 因此必须选择那些对所研究的分类问题有用的量. 其次即使是有用的信息, 有的还不能反映样本的类别特征, 往往要通过某些变换才能得到便于对样本分类的物理量. 这些正是我们前面提到的特征提取的任务. 特征提取方法的基本思想就是, 利用原有的特征构造一批新的特征, 它们是原特征的函数或变换, 但它们更具代表性, 更能反映本质; 同时新特征的总数少于原特征的总数, 实现了特征空间的降维, 却能保留原特征的主要信息. 这一类方法称为降维映射方法.

1.4.1 主成分分析的基本思想

主成分分析是一种常用的线性映射方法, 即用它构造的每个新特征都是原有特征的线性函数. 线性变换相当于坐标系的平移和旋转变换.

我们从直观的例子来说明主成分分析的基本思想. 假设有一个二维数据表, 数据点的分布如图 1.3(a) 所示, 呈椭圆形, 重心为 g. 椭圆的长轴和短轴用 u_1 和 u_2

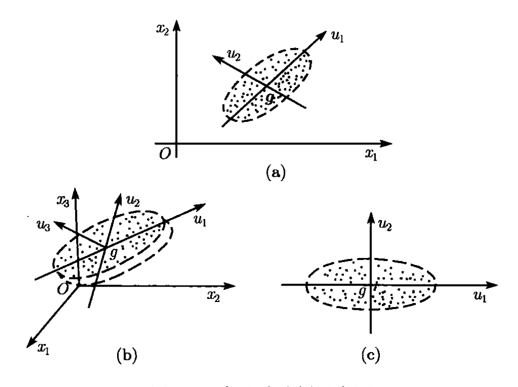


图 1.3 主成分分析示意图

(a) 二维数据的降维; (b) 三维数据的降维; (c) 三维数据降维后的二维投影

表示. 显然, 沿 u_1 方向, 数据的离差最大, 所反映的数据样本总体的信息也最多, 该方向称为样本总体的最大变异方向. 相应地, u_2 是样本总体的次大变异方向. 如果将原点平移到 g, 并且作旋转变换, 便得到一个正交坐标系 u_1gu_2. 可以看出, 若省略 u_2 轴, 将数据点在 u_1 轴上投影, 就得到一个简化的一维数据样本点集. 因此降维处理的核心思想, 就是省却变异较小的变量方向.

再如一个三维数据点集的分布呈椭圆饼形, 如图 1.3(b) 所示, 变异较大的方向为 u_1 和 u_2, 而 u_3 方向的变异很小 (离差很小), 这样若以坐标系 u_1gu_2 来分析数据, 与用原三维空间的数据进行分析, 对结果的差别就会很小.

推广到 n 维的一般情形, 原数据样本点集的特征向量为 $x=(x_1,\cdots,x_n)^{\mathrm{T}}$, 主成分分析实质上是通过坐标系的平移和旋转变换, 使得新坐标系 $\{u_1,\cdots,u_n\}$ 的原点与数据样本点集的重心重合, 各坐标轴 $u_j, j=1,2,\cdots,n$ 之间相互正交, 第一主轴 u_1 是样本总体的最大变异方向, 第二主轴 u_2 是样本总体的次大变异方向, 依此类推. 原数据样本点集在第一主轴 u_1 上的投影值, 构成新数据点集的第一个变量 y_1 称为第一主成分, 依此类推有第 j 主成分 $y_j, j=1,2,\cdots,n$. 主成分分析的结果是

$$\begin{cases} E(y_j)=0, \qquad j=1,2,\cdots,n; \\ V(y_1)\geqslant V(y_2)\geqslant\cdots\geqslant V(y_n). \end{cases} \tag{1.4.1}$$

这样就构成了原数据点集的新的特征向量 $\boldsymbol{y}=(y_1,\cdots,y_n)^{\mathrm{T}}$, 它的各个变量 $y_j, y_k, j\neq k$ 相互之间是互不相关的 (相关系数为 0).

1.4.2 主成分分析算法

假设数据集有 N 个样本点, 原特征向量为 $\boldsymbol{x}=(x_1,\cdots,x_n)^{\mathrm{T}}$, 而新的特征向量为 $\boldsymbol{y}=(y_1,\cdots,y_n)^{\mathrm{T}}$, 每个新特征应是原有特征的线性组合, 即

$$y_j=\boldsymbol{u}_j^{\mathrm{T}}(\boldsymbol{x}-\bar{\boldsymbol{x}})=\sum_{k=1}^{n}u_{jk}(x_k-\bar{x}_k), \qquad j=1,2,\cdots,n. \tag{1.4.2}$$

式中, $\boldsymbol{u}_j=(u_{j1},u_{j2},\cdots,u_{jn})^{\mathrm{T}}$, u_{jk} 是常系数; $\bar{\boldsymbol{x}}=(\bar{x}_1,\cdots,\bar{x}_n)^{\mathrm{T}}$, $\bar{x}_j, j=1,\cdots,n$ 是 N 个样本点第 j 个原特征变量 x_j 的均值 (参见式 (1.3.4)). 写成矩阵的形式即为

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}=\begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ u_{21} & u_{22} & \cdots & u_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ u_{n1} & u_{n2} & \cdots & u_{nn} \end{pmatrix}\begin{pmatrix} x_1-\bar{x}_1 \\ x_2-\bar{x}_2 \\ \cdots \\ x_n-\bar{x}_n \end{pmatrix}$$

或

$$\boldsymbol{y}=\boldsymbol{U}(\boldsymbol{x}-\bar{\boldsymbol{x}}). \tag{1.4.3}$$

问题是怎样求得满足式 (1.4.1) 的 $n\times n$ 矩阵 $\boldsymbol{U}$ 的所有元素呢?

假定训练样本集 N 个样本点的协方差矩阵 $\boldsymbol{V}(\boldsymbol{x})$ 已经算出 (计算公式见式 (1.3.5)~(1.3.7)), 括号内的 $\boldsymbol{x}$ 表示它是用原特征向量计算的. $\boldsymbol{V}(\boldsymbol{x})$ 是一 $n\times n$ 实数对称方阵, 其 n 个特征值 $\lambda_1,\lambda_2,\cdots,\lambda_n$ 及其对应的特征向量 $\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_n$ 可由求解线性齐次方程组

$$[\boldsymbol{V}(\boldsymbol{x})-\lambda_j\boldsymbol{I}]\,\boldsymbol{u}_j=0,\qquad j=1,2,\cdots,n \tag{1.4.4}$$

得到. 由矩阵代数知道, 实数对称方阵的不同特征值 λ_j 对应的特征向量 $\boldsymbol{u}_j$ 是相互正交的, 即

$$\begin{cases}\boldsymbol{u}_j^{\mathrm{T}}\boldsymbol{u}_k=0, & j,k=1,2,\cdots,n, \qquad j\neq k;\\ \boldsymbol{u}_j^{\mathrm{T}}\boldsymbol{u}_k=1, & j=k=1,2,\cdots,n.\end{cases} \tag{1.4.5}$$

不失一般性, 可要求 $\lambda_1\geqslant\lambda_2\geqslant\cdots\geqslant\lambda_n$. 对于确定的协方差矩阵 $\boldsymbol{V}(\boldsymbol{x})$, 特征值 λ_j 及其对应的特征向量 $\boldsymbol{u}_j$ 是唯一确定的.

现在来证明, 这样求得的特征向量 $\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_n$ 和式 (1.4.2) 或式 (1.4.3) 求得的新的特征向量 $\boldsymbol{y}=(y_1,\cdots,y_n)^{\mathrm{T}}$ 均满足式 (1.4.1) 的要求. 利用随机变量的均值运算, 我们有

$$E(y_j)=E\left[\sum_{k=1}^{n}u_{jk}(x_k-\bar{x}_k)\right].$$

注意对随机变量 x_k 的均值运算而言 u_{jk} 是常数, 于是

$$\begin{aligned}E(y_j)&=E\left[\sum_{k=1}^{n}u_{jk}(x_k-\bar{x}_k)\right]=E\left[\sum_{k=1}^{n}u_{jk}x_k\right]-E\left[\sum_{k=1}^{n}u_{jk}\bar{x}_k\right]\\&=\sum_{k=1}^{n}u_{jk}E(x_k)-\sum_{k=1}^{n}u_{jk}E(\bar{x}_k)=0.\end{aligned} \tag{1.4.6}$$

即新的特征向量 $\boldsymbol{y}=(y_1,\cdots,y_n)^{\mathrm{T}}$ 各变量的均值为 0. 利用随机变量的协方差矩阵运算, 当特征向量 $\boldsymbol{y}$ 具有式 (1.4.3) 的形式, 我们有

$$\boldsymbol{V}(\boldsymbol{y})=\boldsymbol{U}\boldsymbol{V}(\boldsymbol{x})\boldsymbol{U}^{\mathrm{T}}.$$

即

$$V_{lm}(\boldsymbol{y})=\sum_{j=1}^{n}\sum_{k=1}^{n}u_{lj}u_{km}V_{jk}(\boldsymbol{x}),\qquad l,m=1,2,\cdots,n.$$

注意到 $\boldsymbol{U}^{\mathrm{T}}$ 的每一列恰好是 $\boldsymbol{V}(\boldsymbol{x})$ 的特征向量并利用条件式 (1.4.4) 可得

$$\boldsymbol{V}(\boldsymbol{x})\boldsymbol{U}^{\mathrm{T}}=\boldsymbol{U}^{\mathrm{T}}\begin{pmatrix}\lambda_1&&&\\&\lambda_2&&\\&&\ddots&\\&&&\lambda_n\end{pmatrix},$$

再由特征向量 $\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_n$ 相互之间的正交性知

$$\boldsymbol{V}(\boldsymbol{y})=\boldsymbol{U}\boldsymbol{V}(\boldsymbol{x})\boldsymbol{U}^{\mathrm{T}}=\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}\begin{pmatrix}\lambda_1&&&\\&\lambda_2&&\\&&\ddots&\\&&&\lambda_n\end{pmatrix}=\begin{pmatrix}\lambda_1&&&\\&\lambda_2&&\\&&\ddots&\\&&&\lambda_n\end{pmatrix}.$$

即

$$\begin{cases}V_{kk}(\boldsymbol{y})=\lambda_k, & \qquad\qquad k=1,2,\cdots,n,\\ V_{kl}(\boldsymbol{y})=0, \quad k\neq l, & \qquad\qquad k,l=1,2,\cdots,n.\end{cases}\tag{1.4.7}$$

因此 $\boldsymbol{y}$ 各变量之间互不相关且相互正交, 它们的方差等于 $\boldsymbol{V}(\boldsymbol{x})$ 的特征值, 且有 $V(y_1)\geqslant V(y_2)\geqslant\cdots\geqslant V(y_n)$. 由此我们证明了式 (1.4.1) 的正确性.

这里顺便提一下 $\boldsymbol{V}(\boldsymbol{x})$ 和 $\boldsymbol{V}(\boldsymbol{y})$ 的一个有用的性质. 由矩阵代数知 $\boldsymbol{n}$ 阶方阵 $\boldsymbol{V}(\boldsymbol{x})$ 的 n 个特征值 $\lambda_1,\lambda_2,\cdots,\lambda_n$ 之和等于 $\boldsymbol{V}(\boldsymbol{x})$ 的迹, 因此有

$$\sum_{j=1}^{n}V_{jj}(\boldsymbol{x})=\sum_{j=1}^{n}V_{jj}(\boldsymbol{y})=\sum_{i=1}^{n}\lambda,\tag{1.4.8}$$

即主成分分析并不改变协方差矩阵对角元素之和.

1.4.3 降维处理及信息损失

由上述叙述可见, 如果在最后几个主轴上各样本点的数值很接近 (离差很小), 亦即新特征向量 $\boldsymbol{y}$ 的最后几个分量 $y_n,y_{n-1},\cdots,y_{p+1}$ 的方差 $\lambda_n,\lambda_{n-1},\cdots,\lambda_{p+1}$ 的数值很小, 它们对于样本点的分类的作用就很小, 略去它们对于样本点的正确分类影响就很小, 因此就可以用 p 维向量 $\hat{\boldsymbol{y}}=(y_1,\cdots,y_p)^{\mathrm{T}}(p\leqslant n)$ 来设计分类器实现样本的分类, 这就是主成分分析的降维处理. 降维处理肯定要丢失样本点集的信息, 我们的目标当然是应当采用信息损失尽可能少的降维方式. 这就要求能对降维处理导致的信息损失作定量的估计.

我们可以定义 $\boldsymbol{y}$ 的第 j 个主成分 y_j 的方差贡献率为

$$r_j=\lambda_j\Big/\sum_{k=1}^{n}\lambda_k,\tag{1.4.9}$$

于是 $\boldsymbol{y}$ 的 n 个成分的累计方差率为 1. 由式 (1.4.8) 知 $\boldsymbol{x}$ 的 n 个成分的累计方差率亦为 1. 当前 p 个主成分的方差贡献率足够接近 1, 就可以只取前 p 个主成分作为新特征. 这时, 降维向量 $\hat{\boldsymbol{y}}$ 的累计方差率为

$$r_{\hat{y}} = \sum_{j=1}^{p} \lambda_j \Big/ \sum_{j=1}^{n} \lambda_j. \tag{1.4.10}$$

因为数据信息主要反映在变量的方差上, 方差越大, 数据包含的信息就越多. 因此, $1 - r_{\hat{y}}$ 可以作为降维处理信息损失的一种度量.

由于主成分分析可以将原来各个变量相互关联的特征向量 $\boldsymbol{x} = (x_1, \cdots, x_n)^{\mathrm{T}}$ 变换为各个变量互不关联的新特征向量 $\boldsymbol{y} = (y_1, \cdots, y_n)^{\mathrm{T}}$, 然后再进一步降维为特征向量 $\hat{\boldsymbol{y}} = (y_1, \cdots, y_p)^{\mathrm{T}} (p \leqslant n)$ 而不带来多少信息损失, 这些性质使得它应用于多级分类器如决策树方法, 特别是超长方体分割法, 能够有效地提高对信号样本的识别效率, 减小计算量. 这一点会在 4.1 节的讨论中加以叙述.

主成分分析带来的问题是主成分的物理含义的解释. 原特征向量 $\boldsymbol{x} = (x_1, \cdots, x_n)^{\mathrm{T}}$ 的各个变量 $x_j (j = 1, 2, \cdots, n)$ 每一个都具有明确的物理意义. 对它们作变换后得到的新综合变量 $y_j (j = 1, 2, \cdots, n)$, 它们的物理意义是什么呢? 可以证明, 如果 $\boldsymbol{x} = (x_1, \cdots, x_n)^{\mathrm{T}}$ 是中心化的, 那么 $\boldsymbol{x}$ 与变换后的 $\boldsymbol{y}$ 之间的相关系数为

$$r(y_j, x_k) = \sqrt{\lambda_j} u_{jk}. \tag{1.4.11}$$

由式 (1.4.2) 知, 对于中心化的 $\boldsymbol{x} = (x_1, \cdots, x_n)^{\mathrm{T}}$, 有

$$y_j = \boldsymbol{u}_j^{\mathrm{T}} \boldsymbol{x} = \sum_{k=1}^{n} u_{jk} x_k, \qquad j = 1, 2, \cdots, n. \tag{1.4.12}$$

可见 y_j 是 n 个变量 $x_k (k = 1, 2, \cdots, n)$ 的线性组合, 组合系数正比于 y_j 与 x_k 之间的相关系数 $r(y_j, x_k)$. 因此人们可以通过观察组合系数 u_{jk} 的符号和大小, 对 y_j 的物理含义作出判断. 正的 u_{jk} 说明 y_j 与 x_k 之间正相关, 大的 u_{jk} 值说明 y_j 与 x_k 之间关联强, 反之则关联弱.

第二章　贝叶斯决策

模式识别的分类问题是根据待识别样本的特征向量的观测值将样本归之为某个类别. 统计决策理论是处理模式分类的基本统计理论之一, 它对模式分析和分类器的设计有指导意义. 贝叶斯 (Bayes) 决策理论是统计模式识别中的一个基本方法, 为此我们首先对于贝叶斯决策和利用它进行模式识别的一些问题作简要的介绍.

利用贝叶斯决策方法进行样本分类时有两个前提条件:

(1) 要决策分类的类别数是一定的;

(2) 各类别的总体概率分布是已知的.

假定要决策分类的类别数用 c 表示, 各类别的状态用 ω_i 表示, $i=1,2,\cdots,c$. 假定要识别的物理样本有 n 个特征观测量 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$, 即样本为 n 维特征向量, 由于每个特征观测量都是随机变量, 所以 $\boldsymbol{x}$ 是 n 维随机变量. 它的每一个观测值可以看成是 n 维特征空间中的一个点. 前提条件 (1) 要求 c 为已知常数, 条件 (2) 要求对应于各类别 ω_i 出现的先验概率 $\pi(\omega_i)$ 是已知的, 并且当样本 $\boldsymbol{x}\in\omega_i$ 时的条件概率密度 $p(\boldsymbol{x}|\omega_i)$ 也是已知的.

于是贝叶斯决策分类要解决的问题归结为, 对于一个特定的 $\boldsymbol{x}$ 样本, 在满足以上两个前提条件的情况下, 怎样对其归类.

2.1　基于最小错误率的贝叶斯决策

2.1.1　决策规则

模式分类的重要要求之一是尽量降低对样本错误分类的比率. 利用贝叶斯公式, 能得到错分率最小的分类规则, 称为基于最小错误率的贝叶斯决策. 当先验概率 $\pi(\omega_i)$ 和随机变量 $\boldsymbol{x}\in\omega_i$ 的条件概率密度 $p(\boldsymbol{x}|\omega_i)$ 均为已知时, 利用贝叶斯公式可求得所谓的后验概率 $q(\omega_i|\boldsymbol{x})$:

$$q(\omega_i|\boldsymbol{x})=\frac{p(\boldsymbol{x}|\omega_i)\pi(\omega_i)}{\sum\limits_{j=1}^{c}p(\boldsymbol{x}|\omega_j)\pi(\omega_j)},\qquad i=1,2,\cdots,c. \tag{2.1.1}$$

后验概率综合了先验概率 $\pi(\omega_i)$ 和样本测量值 $\boldsymbol{x}$ 对于样本属于各类别的状态 ω_i 的概率大小的新知识, 也就是综合了随机试验前的先验知识 $\pi(\omega_i)$ 和随机试验的知识

(随机变量 $\boldsymbol{x}$ 的一次随机测量值), 贝叶斯统计认为后验概率 $q(\omega_i|\boldsymbol{x})$ 是统计决策的基础.

基于最小错误率的贝叶斯决策规则为

$$\boldsymbol{x} \in \omega_i, \qquad \text{当 } q(\omega_i|\boldsymbol{x}) = \max_{j=1,\cdots,c} q(\omega_j|\boldsymbol{x}) \text{ 时}. \tag{2.1.2}$$

还可以得到基于最小错误率的贝叶斯决策规则的等价形式:

$$\boldsymbol{x} \in \omega_i, \qquad \text{当 } p(\boldsymbol{x}|\omega_i)\pi(\omega_i) = \max_{j=1,\cdots,c} p(\boldsymbol{x}|\omega_j)\pi(\omega_j) \text{ 时}, \tag{2.1.3}$$

$$\boldsymbol{x} \in \omega_i, \qquad \text{当 } l(\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_i)}{p(\boldsymbol{x}|\omega_j)} > \frac{\pi(\omega_j)}{\pi(\omega_i)}, j = 1, \cdots, c \text{ 且 } j \neq i \text{ 时}, \tag{2.1.4}$$

$$\boldsymbol{x} \in \omega_i, \qquad \text{当 } \ln p(\boldsymbol{x}|\omega_i) + \ln \pi(\omega_i) > \ln p(\boldsymbol{x}|\omega_j) + \ln \pi(\omega_j), \quad j = 1, \cdots, c \text{ 且 } j \neq i \text{ 时}. \tag{2.1.5}$$

对于最简单的 $c = 2$ 类问题, 则有

$$\boldsymbol{x} \in \omega_i, \qquad \text{当 } q(\omega_i|\boldsymbol{x}) = \max_{j=1,2} q(\omega_j|\boldsymbol{x}) \text{ 时}, \tag{2.1.6}$$

以及等价形式:

$$\boldsymbol{x} \in \omega_i, \qquad \text{当 } p(\boldsymbol{x}|\omega_i)\pi(\omega_i) = \max_{j=1,2} p(\boldsymbol{x}|\omega_j)\pi(\omega_j) \text{ 时}. \tag{2.1.7}$$

定义似然比

$$l(\boldsymbol{x}) \equiv \frac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)}.$$

则有

$$\begin{cases} \boldsymbol{x} \in \omega_1, & \text{当 } l(\boldsymbol{x}) > \pi(\omega_2)/\pi(\omega_1) \text{ 时}, \\ \boldsymbol{x} \in \omega_2, & \text{当 } l(\boldsymbol{x}) < \pi(\omega_2)/\pi(\omega_1) \text{ 时}, \end{cases} \tag{2.1.8}$$

以及

$$\begin{cases} \boldsymbol{x} \in \omega_1, & \text{当 } h(\boldsymbol{x}) < \ln\left[\dfrac{\pi(\omega_1)}{\pi(\omega_2)}\right] \text{ 时}, \\[2ex] \boldsymbol{x} \in \omega_2, & \text{当 } h(\boldsymbol{x}) > \ln\left[\dfrac{\pi(\omega_1)}{\pi(\omega_2)}\right] \text{ 时}. \end{cases} \tag{2.1.9}$$

式中

$$h(\boldsymbol{x}) \equiv -\ln[l(\boldsymbol{x})] = -\ln p(\boldsymbol{x}|\omega_1) + \ln p(\boldsymbol{x}|\omega_2).$$

式中的 $\pi(\omega_2)/\pi(\omega_1)$ 称为似然比阈值.

2.1.2 错误率

现在来讨论错误率问题. 我们从比较简单的二类问题出发, 并假定 $n=1$ 即特征空间为一维. 一个二类问题的后验概率假定如图 2.1 所示. 由式 (2.1.6) 知, 若 $q(\omega_1|\boldsymbol{x}) > q(\omega_2|\boldsymbol{x})$, 则样本 $\boldsymbol{x}$ 决策为 ω_1 类; 但如图 2.1 可知, 这时仍有概率 $q(\omega_2|\boldsymbol{x})$ 样本 $\boldsymbol{x}$ 属于 ω_2 类. 因此样本 $\boldsymbol{x}$ 决策为 ω_1 类的条件错误概率为 $q(\omega_2|\boldsymbol{x})$. 类似地, 样本 $\boldsymbol{x}$ 决策为 ω_2 类的条件错误概率为 $q(\omega_1|\boldsymbol{x})$. 于是样本 $\boldsymbol{x}$ 决策的条件错误概率 $\varepsilon(e|\boldsymbol{x})$ 可表示为

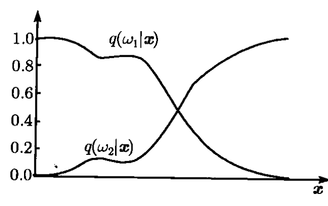


图 2.1 后验概率

$$\varepsilon(e|\boldsymbol{x}) = \begin{cases} q(\omega_1|\boldsymbol{x}), & \text{当 } q(\omega_2|\boldsymbol{x}) > q(\omega_1|\boldsymbol{x}) \text{ 时}, \\ q(\omega_2|\boldsymbol{x}), & \text{当 } q(\omega_1|\boldsymbol{x}) > q(\omega_2|\boldsymbol{x}) \text{ 时}. \end{cases} \tag{2.1.10}$$

上式也可写成其等价的形式

$$\varepsilon(e|\boldsymbol{x}) = \min\left[q(\omega_1|\boldsymbol{x}), q(\omega_2|\boldsymbol{x})\right]. \tag{2.1.11}$$

平均错误率定义为

$$\varepsilon(e) = \int_{-\infty}^{\infty} \varepsilon(e, \boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \int_{-\infty}^{\infty} \varepsilon(e|\boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}. \tag{2.1.12}$$

式中积分在 n 维特征空间中进行, $p(\boldsymbol{x})$ 为随机变量 $\boldsymbol{x}$ 的边沿概率. 对于 c 类问题, $p(\boldsymbol{x})$ 的表式为

$$p(\boldsymbol{x}) = \sum_{i=1}^{c} p(\boldsymbol{x}|\omega_i)\pi(\omega_i). \tag{2.1.13}$$

令 t 为两类样本的分界面, 当特征向量为一维时, t 是 x 轴上的一个点, 将 x 轴分为两个区域 $R_1 \in (-\infty, t)$ 和 $R_2 \in (t, \infty)$, 当样本 $\boldsymbol{x} \in R_1(R_2)$ 时判为 $\omega_1(\omega_2)$ 类. 这样平均错误率为

$$\begin{aligned} \varepsilon(e) &= \int_{-\infty}^{t} q(\omega_2|\boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} + \int_{t}^{\infty} q(\omega_1|\boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ &= \int_{-\infty}^{t} p(\boldsymbol{x}|\omega_2)\pi(\omega_2) \mathrm{d}\boldsymbol{x} + \int_{t}^{\infty} p(\boldsymbol{x}|\omega_1)\pi(\omega_1) \mathrm{d}\boldsymbol{x}. \end{aligned}$$

由于 $\pi(\omega_1),\pi(\omega_2)$ 是已知常量, 可以提出积分号外, 上式可进一步写为

$$\begin{aligned}\varepsilon(e) &= \pi(\omega_2)\int_{-\infty}^{t} p(\boldsymbol{x}|\omega_2)\mathrm{d}\boldsymbol{x} + \pi(\omega_1)\int_{t}^{\infty} p(\boldsymbol{x}|\omega_1)\mathrm{d}\boldsymbol{x} \\ &= \pi(\omega_2)\int_{R_1} p(\boldsymbol{x}|\omega_2)\mathrm{d}\boldsymbol{x} + \pi(\omega_1)\int_{R_2} p(\boldsymbol{x}|\omega_1)\mathrm{d}\boldsymbol{x} \\ &\equiv \pi(\omega_2)\varepsilon_{12}(e) + \pi(\omega_1)\varepsilon_{21}(e). \end{aligned} \tag{2.1.14}$$

式中

$$\varepsilon_{21}(e) = \int_{R_2} p(\boldsymbol{x}|\omega_1)\mathrm{d}\boldsymbol{x} \tag{2.1.15}$$

表示 ω_1 类样本 $\boldsymbol{x}$ 落在 $R_2 \in (t,\infty)$ 区域被决策为 ω_2 类时的错误概率, 类似地, $\varepsilon_{12}(e) = \int_{R_1} p(\boldsymbol{x}|\omega_2)\mathrm{d}\boldsymbol{x}$ 表示 ω_2 类样本 $\boldsymbol{x}$ 落在 $R_1 \in (-\infty,t)$ 区域被决策为 ω_1 类时的错误概率. 图 2.2 中的网格线和斜线区域的面积即为 $\pi(\omega_1)\varepsilon_{21}(e)$ 和 $\pi(\omega_2)\varepsilon_{12}(e)$. 以上讨论不难推广到 n 维特征空间的情形.

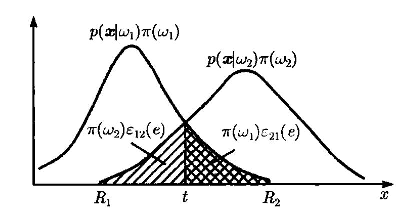


图 2.2　错误率

从式 (2.1.10) 知道, 决策规则式 (2.1.2), (2.1.6) 实际上是使样本 $\boldsymbol{x}$ 决策的条件错误概率 $q(e|\boldsymbol{x})$ 取小者, 这就使式 (2.1.12) 定义的平均错误率 $\varepsilon(e)$ 达到最小. 这就证明了基于最小错误率的贝叶斯决策规则确实使平均错误率 $\varepsilon(e)$ 达到最小.

多类决策问题中, 特征空间被分成 $R_1,R_2,\cdots,R_c$ 个区域, 可能错分的情况很多, 平均错误概率 $\varepsilon(e)$ 将由 $c(c-1)$ 项组成, 即

$$\begin{aligned}\varepsilon(e) &= \left.\begin{aligned}&\pi(\omega_1)\left[\varepsilon(\boldsymbol{x}\in R_2|\omega_1) + \varepsilon(\boldsymbol{x}\in R_3|\omega_1) + \cdots + \varepsilon(\boldsymbol{x}\in R_c|\omega_1)\right] \\ &+\pi(\omega_2)\left[\varepsilon(\boldsymbol{x}\in R_1|\omega_2) + \varepsilon(\boldsymbol{x}\in R_3|\omega_2) + \cdots + \varepsilon(\boldsymbol{x}\in R_c|\omega_2)\right] \\ &\qquad\cdots \\ &\pi(\omega_c)\left[\varepsilon(\boldsymbol{x}\in R_1|\omega_c) + \varepsilon(\boldsymbol{x}\in R_2|\omega_c) + \cdots + \varepsilon(\boldsymbol{x}\in R_{c-1}|\omega_c)\right]\end{aligned}\right\} c\text{ 行} \\ &= \sum_{i=1}^{c}\sum_{\substack{j=1\\ j\neq i}}^{c}\pi(\omega_i)\left[\varepsilon(\boldsymbol{x}\in R_j|\omega_i)\right] = \sum_{i=1}^{c}\sum_{\substack{j=1\\ j\neq i}}^{c}\pi(\omega_i)\varepsilon_{ji}(e). \end{aligned} \tag{2.1.16}$$

式中, $\varepsilon(\boldsymbol{x} \in R_j|\omega_i) = \varepsilon_{ji}(e)$ 是 ω_i 类样本但其 $\boldsymbol{x}$ 落在 R_j 区域因而被决策为 ω_j 类的错误概率. 由上式可知直接求 $\varepsilon(e)$ 的计算量很大. 我们可以计算平均的正确分类概率 $\varepsilon(c)$, 则平均错误分类概率 $\varepsilon(e)$ 为

$$\varepsilon(e) = 1 - \varepsilon(c), \tag{2.1.17}$$

因为 $\varepsilon(\boldsymbol{x} \in R_j|\omega_j)$ 是 ω_j 类样本但其 $\boldsymbol{x}$ 值落在 R_j 区域因而被正确决策为 ω_j 类的概率, 所以对于所有 c 类样本总的正确分类概率的期望值 $\varepsilon(c)$ 由下式求得:

$$\begin{aligned}\varepsilon(c) &= \sum_{j=1}^{c} \pi(\omega_j)\varepsilon(\boldsymbol{x} \in R_j|\omega_j) = \sum_{j=1}^{c} \pi(\omega_j)\varepsilon_{jj} \\ &= \sum_{j=1}^{c} \int_{R_j} p(\boldsymbol{x}|\omega_j)\pi(\omega_j)\mathrm{d}\boldsymbol{x}.\end{aligned} \tag{2.1.18}$$

式中求和号内只有 c 项, 比直接求 $\varepsilon(e)$ 容易得多.

2.1.3 分类器设计

基于以上最小错误率的贝叶斯决策的分析, 可以进行相应的分类器设计.

1. c 类情形

对于 c 类分类问题, 按照决策规则可以把 n 维特征空间分成 c 个决策域, 定义一组判别函数 $g_i(\boldsymbol{x}),i = 1, 2, \cdots, c$, 决策规则为: 若 $g_i(\boldsymbol{x}) > g_j(\boldsymbol{x})$ 对一切 $j \neq i$ 成立, 则将 $\boldsymbol{x}$ 归为 ω_i 类. 按照式 (2.1.2)~(2.1.5) 的决策规则, 显然这里判别函数 $g_i(\boldsymbol{x})$ 可定义为

$$\begin{cases} g_i(\boldsymbol{x}) = q(\omega_i|\boldsymbol{x}) \\ g_i(\boldsymbol{x}) = p(\boldsymbol{x}|\omega_i)\pi(\omega_i) \\ g_i(\boldsymbol{x}) = \ln p(\boldsymbol{x}|\omega_i) + \ln \pi(\omega_i) \end{cases}. \tag{2.1.19}$$

各决策域 R_i 被决策面所分割, 这些决策面是特征空间中的超曲面, 相邻的两个决策域在决策面上其判别函数值是相等的, 即如 R_i 和 R_j 相邻, 则分割它们的决策面方程为

$$g_i(\boldsymbol{x}) = g_j(\boldsymbol{x}). \tag{2.1.20}$$

根据以上原则, 可以编写基于贝叶斯决策的分类器的计算机软件. 它的功能是先计算出 c 个判别函数 $g_i(\boldsymbol{x})$, 从中选出对应于判别函数为最大值的类作为决策结果. 图 2.3 是这种分类器的示意图.

2. 二类情形

对于二类问题, 只需要定义一个判别函数 $g(\boldsymbol{x})$,

$$g(\boldsymbol{x}) = g_1(\boldsymbol{x}) - g_2(\boldsymbol{x}) \tag{2.1.21}$$

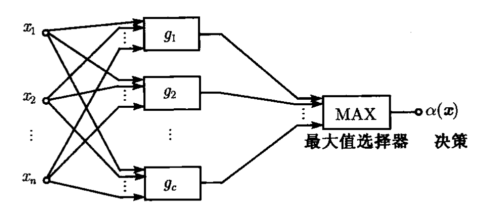


图 2.3　多类分类器示意图

决策规则为：若 $g(\boldsymbol{x})>0$, 则将 $\boldsymbol{x}$ 归为 ω_1 类; 若 $g(\boldsymbol{x})<0$, 则将 $\boldsymbol{x}$ 归为 ω_2 类. 按照式 (2.1.6)~(2.1.9) 的决策规则, 显然这里判别函数 $g(\boldsymbol{x})$ 可定义为

$$\begin{cases} g(\boldsymbol{x})=q(\omega_1|\boldsymbol{x})-q(\omega_2|\boldsymbol{x}) \\ g(\boldsymbol{x})=p(\boldsymbol{x}|\omega_1)\pi(\omega_1)-p(\boldsymbol{x}|\omega_2)\pi(\omega_2) \\ g(\boldsymbol{x})=\ln\dfrac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)}-\ln\dfrac{\pi(\omega_1)}{\pi(\omega_2)} \end{cases}. \tag{2.1.22}$$

决策域 R_1 和 R_2 被决策面所分割, 决策面方程为

$$g(\boldsymbol{x})=0. \tag{2.1.23}$$

一般地说, $\boldsymbol{x}$ 为一维时, 决策面为一分界点; $\boldsymbol{x}$ 为二维时, 决策面为一分界曲线; $\boldsymbol{x}$ 为三维时, 决策面为一分界曲面; $\boldsymbol{x}$ 为 c 维 $(c>3)$ 时, 决策面为一分界超曲面.

二类分类器先计算出判别函数 $g(\boldsymbol{x})$, 根据其正负对 $\boldsymbol{x}$ 进行分类. 图 2.4 是这种分类器的示意图.

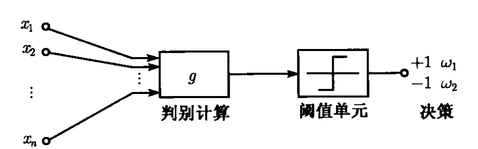


图 2.4　二类分类器示意图

2.2　Neyman-Pearson 决策

在两类问题的决策中, 存在两种错判的可能性：样本属于 ω_1 类但被分类器决策为 ω_2 类, 以及样本属于 ω_2 类但被分类器决策为 ω_1 类, 这两种错判的概率分别为 $\pi(\omega_1)\varepsilon_{21}(e)$ 和 $\pi(\omega_2)\varepsilon_{12}(e)$. 实际问题中, 先验概率 $\pi(\omega_1)$ 和 $\pi(\omega_2)$ 往往是确定的,

所以 $\varepsilon_{21}(e)$ 和 $\varepsilon_{12}(e)$ 一般称为两类错误率. 最小错误率贝叶斯决策是使 $\pi(\omega_1)\varepsilon_{21}(e)$ 和 $\pi(\omega_2)\varepsilon_{12}(e)$ 之和达到最小 (参见式 (2.1.14)):

$$\varepsilon(e) = \pi(\omega_2)\varepsilon_{12}(e) + \pi(\omega_1)\varepsilon_{21}(e)$$

但在实际问题中, 有时要求其中的一类错误率不得大于某个给定常数而使另一类错误率尽可能地小. 如在对病人进行癌细胞检查时, 显然, 把癌细胞误判为正常细胞会导致严重的后果, 因此要求这种误判率很小, 即要求 $\varepsilon_{12}(e) = \varepsilon_0$, ε_0 为一很小的常数, 在这种条件下再要求 $\varepsilon_{21}(e)$ 尽可能地小.

这类决策问题是在 $\varepsilon_{12}(e) = \varepsilon_0$ 条件下求 $\varepsilon_{21}(e)$ 极小值的约束极值问题, 可以用拉格朗日乘子法求解. 引入拉格朗日乘子 λ, 定义量 γ 为

$$\gamma = \varepsilon_{21}(e) + \lambda\left(\varepsilon_{12}(e) - \varepsilon_0\right) \tag{2.2.1}$$

这样, 使 γ 达到极小的 λ 值 λ^* 对应于问题的解.

由式 (2.1.14) 可知

$$\varepsilon_{21}(e) = \int_{R_2} p(\boldsymbol{x}|\omega_1)\mathrm{d}\boldsymbol{x} \tag{2.2.2}$$

$$\varepsilon_{12}(e) = \int_{R_1} p(\boldsymbol{x}|\omega_2)\mathrm{d}\boldsymbol{x}, \tag{2.2.3}$$

其中, R_1 是类别 ω_1 的决策域; R_2 是类别 ω_2 的决策域; $R = R_1 + R_2$ 为整个特征空间. 决策是将整个特征空间分割成互不相交的两个区域 R_1 和 R_2. 两个区域的分界点 (面) 令为 t. 若待分类样本 $\boldsymbol{x}$ 落入区域 R_1, 则样本归类为 ω_1, 反之样本归类为 ω_2. 根据条件概率密度的性质, 有

$$\int_{R_2} p(\boldsymbol{x}|\omega_1)\mathrm{d}\boldsymbol{x} = 1 - \int_{R_1} p(\boldsymbol{x}|\omega_1)\mathrm{d}\boldsymbol{x} \tag{2.2.4}$$

将式 (2.2.2) 和式 (2.2.3) 代入式 (2.2.1), 并考虑到式 (2.2.4) 可得

$$\begin{aligned}\gamma &= \int_{R_2} p(\boldsymbol{x}|\omega_1)\mathrm{d}\boldsymbol{x} + \lambda\left[\int_{R_1} p(\boldsymbol{x}|\omega_2)\mathrm{d}\boldsymbol{x} - \varepsilon_0\right] \\ &= (1 - \lambda\varepsilon_0) + \int_{R_1} [\lambda p(\boldsymbol{x}|\omega_2) - p(\boldsymbol{x}|\omega_1)]\mathrm{d}\boldsymbol{x}\end{aligned} \tag{2.2.5}$$

将上式分别对分界点 (面) t 和参数 λ 求导, 并令其等于零, 即 $\dfrac{\partial\gamma}{\partial t} = 0$ 及 $\dfrac{\partial\gamma}{\partial\lambda} = 0$, 则可得

$$\lambda^* = \frac{p(\boldsymbol{t}|\omega_1)}{p(\boldsymbol{t}|\omega_2)} \tag{2.2.6}$$

$$\int_{R_1} p(\boldsymbol{x}|\omega_2)\mathrm{d}\boldsymbol{x} = \varepsilon_0 \tag{2.2.7}$$

可见最佳 λ 值 λ^* 等于分界点处样本 $\boldsymbol{x} \in \omega_1$ 时的条件概率密度 $p(\boldsymbol{x}|\omega_1)$ 和 $\boldsymbol{x} \in \omega_2$ 时的条件概率密度 $p(\boldsymbol{x}|\omega_2)$ 的比值. 满足式 (2.2.6) 的最佳 λ 值和满足式 (2.2.7) 的边界面使 γ 达到极小. 这时决策规则可写为

$$\begin{cases} \dfrac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)} > \lambda^*, & \text{则 } \boldsymbol{x} \in \omega_1; \\ \dfrac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)} < \lambda^*, & \text{则 } \boldsymbol{x} \in \omega_2. \end{cases} \tag{2.2.8}$$

这种限定一类错误率 $\varepsilon_{12}(e) = \varepsilon_0$ 为常数而使另一类错误率 ε_{21} 达到极小的决策规则称为 Neyman-Pearson 决策规则.

回顾最小错误率贝叶斯决策规则式 (2.1.8)

$$\begin{cases} \dfrac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)} > \dfrac{\pi(\omega_2)}{\pi(\omega_1)}, & \text{则 } \boldsymbol{x} \in \omega_1; \\ \dfrac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)} < \dfrac{\pi(\omega_2)}{\pi(\omega_1)}, & \text{则} \boldsymbol{x} \in \omega_2. \end{cases}$$

可见 Neyman-Pearson 决策规则和最小错误率贝叶斯决策规则都是以似然比为基础的, 所不同的是最小错误率贝叶斯决策的阈值是先验概率之比, 而 Neyman-Pearson 决策的阈值是拉格朗日乘子 λ^*, 即两类样本在分界点处的条件概率密度之比.

2.3　正态分布时的贝叶斯决策

利用贝叶斯决策方法进行样本分类的前提条件之一是：各类别的条件概率密度 $p(\boldsymbol{x}|\omega_i)$ 为已知. 用多元正态分布作为各类别的条件概率密度是常用的选择之一, 原因是对于许多实际的数据样本集, 正态性假设通常是一种较合理的近似. 当然如果要用多元正态分布作为类条件概率密度来求得最终的分类结果, 必须注意其物理上的合理性, 即应当先进行假设检验证明该假设确实可用. 否则基于正态性假设求得的分类结果只能视为某种近似.

多元正态分布的概率密度函数为

$$p(\boldsymbol{x}) = N(\boldsymbol{\mu}, \boldsymbol{V}) \equiv \frac{1}{(2\pi)^{n/2}|\boldsymbol{V}|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{V}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\} \tag{2.3.1}$$

式中, $\boldsymbol{x}$ 为 n 维特征向量; $\boldsymbol{\mu}$ 是其 n 维均值向量; $\boldsymbol{V}$ 是 $n \times n$ 阶协方差矩阵; $\boldsymbol{V}^{-1}$ 是其逆矩阵; $|\boldsymbol{V}|$ 是协方差矩阵的行列式.

假定各类别的条件概率密度为多元正态分布, 即

$$p(\boldsymbol{x}|\omega_i) \sim N(\mu_i, \boldsymbol{V}_i), \qquad i = 1, \cdots, c \tag{2.3.2}$$

代入基于最小错误率的贝叶斯决策判别函数式 (2.1.19) 中的对数形式, 立即得到

$$g_i(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}\boldsymbol{V}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i) - \frac{n}{2}\ln 2\pi - \frac{1}{2}\ln|\boldsymbol{V}_i| + \ln\pi(\omega_i)$$

其中右边第二项 $\frac{n}{2}\ln 2\pi$ 与类别 i 无关, 因而可以略去, 从而判别函数可写为

$$g_i(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}\boldsymbol{V}_i^{-1}(x-\boldsymbol{\mu}_i) - \frac{1}{2}\ln|\boldsymbol{V}_i| + \ln\pi(\omega_i) \tag{2.3.3}$$

决策面方程为

$$g_i(\boldsymbol{x}) - g_j(\boldsymbol{x}) = 0. \tag{2.3.4}$$

决策规则为：若 $g_i(\boldsymbol{x}) - g_j(\boldsymbol{x}) > 0$ 对一切 $j \neq i$ 成立, 则将 $\boldsymbol{x}$ 归为 ω_i 类.

为了搞清楚决策面的形状, 将式 (2.3.3) 改写为

$$g_i(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{W}_i\boldsymbol{x} + \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x} + w_{i0} \tag{2.3.5}$$

其中

$$\begin{aligned} \boldsymbol{W}_i &= -\frac{1}{2}\boldsymbol{V}_i^{-1}, \qquad (n \times n \text{ 维矩阵}) \\ \boldsymbol{w}_i &= \boldsymbol{V}_i^{-1}\boldsymbol{\mu}_i \qquad (n \text{ 维列向量}) \\ w_{i0} &= -\frac{1}{2}\boldsymbol{\mu}_i^{\mathrm{T}}\boldsymbol{V}_i^{-1}\boldsymbol{\mu}_i - \frac{1}{2}\ln|\boldsymbol{V}_i| + \ln\pi(\omega_i) \end{aligned} \tag{2.3.6}$$

于是决策面方程式 (2.3.4) 可写成

$$x^{\mathrm{T}}(\boldsymbol{W}_i - \boldsymbol{W}_j)\boldsymbol{x} + (\boldsymbol{w}_i^{\mathrm{T}} - \boldsymbol{w}_j^{\mathrm{T}})\boldsymbol{x} + (w_{i0} - w_{j0}) = 0 \tag{2.3.7}$$

式 (2.3.7) 的决策面方程为 $\boldsymbol{x}$ 的二次型, 对应的决策面为超二次曲面, 随着 $\boldsymbol{V}_i$, $\boldsymbol{\mu}_i$, $\pi(\omega_i)$ 的不同而呈现为某种超二次曲面, 即超椭球面、超球面、超抛物面、超双曲面或超平面.

图 2.5 显示了两类样本的条件概率密度 $p(\boldsymbol{x}|\omega_i), i = 1, 2$ 服从二维正态分布情形下的决策面的不同形式. 在 (a)~(e) 五种形式中, 样本的变量 x_1(横坐标值) 和 x_2(纵坐标值) 之间是相互独立的, 所以协方差矩阵为对角阵. 进一步假定两类的先验概率相等即 $\pi(\omega_1) = \pi(\omega_2)$, 那么决策面的形状完全由 $\boldsymbol{V}_i$, $\boldsymbol{\mu}_i$, $i = 1, 2$ 决定. 图 2.5 中以标号 1,2 的等概率密度轮廓线来表征相应类别样本分布的标准离差. 五种决策面的形状如下：

(a) $\sigma_{x1}(\omega_1) = \sigma_{x2}(\omega_1)$, $\sigma_{x1}(\omega_2) = \sigma_{x2}(\omega_2)$, $\sigma(\omega_1) > \sigma(\omega_2)$. 决策面为圆.

(b) $\sigma_{x1}(\omega_1) < \sigma_{x2}(\omega_1)$, $\sigma_{x1}(\omega_2) < \sigma_{x2}(\omega_2)$, $\sigma_{x1/x2}(\omega_1) > \sigma_{x1/x2}(\omega_2)$. 决策面为椭圆.

(c) $\sigma_{x1}(\omega_1) = \sigma_{x1}(\omega_2) = \sigma_{x2}(\omega_2)$, $\sigma_{x2}(\omega_1) > \sigma_{x2}(\omega_2)$. 决策面为抛物线.

(d) $\sigma_{x1}(\omega_1) = \sigma_{x2}(\omega_2)$, $\sigma_{x2}(\omega_1) = \sigma_{x1}(\omega_2)$. 决策面为双曲线.

(e) 标准离差情况同 (d), 但 μ_1,μ_2 位置有特定的对称性. 决策面为两条直线.

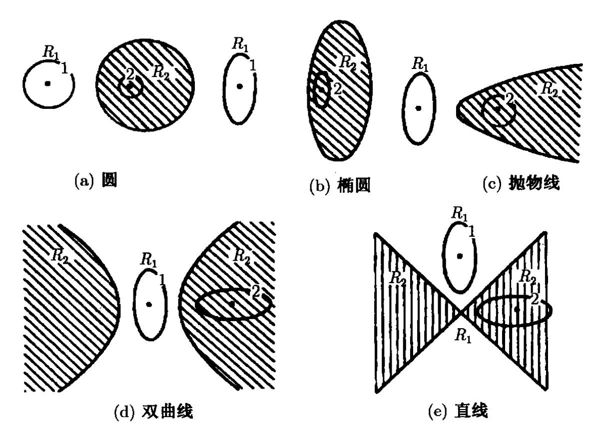


图 2.5　两类样本的条件概率密度服从二维正态分布情形下的决策面

2.4　分类器的效率和错误率

本小节关于效率和错误率的讨论不仅对贝叶斯分类器, 而且对其他分类器都适用.

2.4.1　分类器的效率、错误率和判选率矩阵

假定有 c 个模式类, 用 $\omega_1,\cdots,\omega_c$ 表示. 我们用某种方法设计了一个分类器对任意事例归为这 c 个模式类之一. 一般而言, 一个分类器的性能不大可能是完全理想的, 即既可能把 ω_i 类的样本正确地判别为 ω_i 类, 也可能错误地判别为 $\omega_j(j \neq i)$ 类. 为此我们可定义分类器将一个 ω_j 类的样本判别为一个 ω_i 类的判选率为 ε_{ij}, 即 ε_{ij} 的第一个下标 (i) 标记分类器对样本的判定类别, 第二个下标 (j) 标记样本的真实类别. 于是对于 c 个模式类的情形, 就得到一个 $c \times c$ 的判选率 (或效率) 矩阵 ε:

$$\varepsilon = \begin{pmatrix} \varepsilon_{11} & \varepsilon_{12} & \cdots & \varepsilon_{1c} \\ \varepsilon_{21} & \varepsilon_{22} & \cdots & \varepsilon_{2c} \\ \vdots & \vdots & \varepsilon_{ij} & \vdots \\ \varepsilon_{c1} & \varepsilon_{c2} & \cdots & \varepsilon_{cc} \end{pmatrix} \tag{2.4.1}$$

判选率矩阵 ε 的对角元素 ε_{ii} 表示分类器把 $\omega_i(i=1,2,\cdots,c)$ 类的样本正确地判别为 ω_i 类的概率, 也可称为分类器正确分类的效率; 而非对角元素 ε_{ij} 表示分类器把 $\omega_j(j=1,2,\cdots,c)$ 类的样本错误地判别为其他类别 $\omega_i(i\neq j)$ 的概率, 亦即错判率或错误率. 可见判选率矩阵 ε 表征了分类器的优劣, 因此判选率矩阵 ε 是分类器的一个非常重要的参数. 一个 j 类样本被分类器判为 i 类样本 $(i=1,2,\cdots,c)$ 的概率之和应当等于 1, 即有

$$\sum_{i=1}^{c}\varepsilon_{ij}=1, \qquad j=1,2,\cdots,c \tag{2.4.2}$$

应当指出, 一般情况下 ε 是个非对称矩阵 $\varepsilon_{ij}\neq\varepsilon_{ji},(i\neq j)$, 即把 $\omega_i(i=1,2,\cdots,c)$ 类的样本错判为类别 $\omega_j(j\neq i)$ 的概率不等于把 ω_j 类的样本错判为类别 ω_i 的概率. 一个理想的分类器的效率矩阵 ε 其对角元素皆为 1, 而非对角元素皆为 0. 对于大多数的实际情况, 对于一个好的分类器的要求应当是对角元素尽可能接近 1, 而非对角元素接近 0.

根据最小错误率贝叶斯决策规则下关于分类器正确率和错误率的讨论, 将公式 (2.1.18) 与上述表述对照, 可知最小错误率贝叶斯决策分类器的效率和错误率为

$$\varepsilon(c)=\sum_{j=1}^{c}\pi(\omega_j)\varepsilon_{jj}=\sum_{j=1}^{c}\int_{R_j}p(\boldsymbol{x}|\omega_j)\pi(\omega_j)\mathrm{d}\boldsymbol{x} \tag{2.4.3}$$

$$\varepsilon(e)=1-\varepsilon(c) \tag{2.4.4}$$

把 $\omega_j(j=1,2,\cdots,c)$ 类的样本错误地判别为其他类别 $\omega_i(i\neq j)$ 的错误率为

$$\varepsilon_{ij}=\varepsilon\left(\boldsymbol{x}\in R_i|\omega_j\right)=\int_{R_i}p(\boldsymbol{x}|\omega_j)\mathrm{d}\boldsymbol{x}, \quad i\neq j. \tag{2.4.5}$$

把所有不同于 i 类的样本错误地判别为 ω_i 类的错误率为

$$\varepsilon_i=\sum_{j\neq i,j=1}^{c}\varepsilon_{ij}=\sum_{j\neq i,j=1}^{c}\pi(\omega_j)\int_{R_i}p(\boldsymbol{x}|\omega_j)\mathrm{d}\boldsymbol{x}. \tag{2.4.6}$$

由此可以看到当数据样本 $\boldsymbol{x}$ 为多维向量时, 效率和错误率要进行多重积分计算; 同时, 决策域 $R_1,R_2,\cdots,R_c$ 的确定也是十分困难的. 所以, 虽然效率和错误率的概念比较简单, 但在多维情况下类条件密度的解析表达式比较复杂时, 它们的计算是相当困难的.

在许多实际问题中, 贝叶斯决策的重要前提 (要求各类别 ω_i 出现的先验概率 $\pi(\omega_i)$ 和样本 $\boldsymbol{x}\in\omega_i$ 时的条件概率密度 $p(\boldsymbol{x}|\omega_i)$ 都为已知) 往往不能满足, 因而效率矩阵的解析求解成为不可能.

第一章里我们多次提到, 许多实际问题中涉及的往往是两类样本的分类问题, 即分为信号样本和本底样本. 这时判选率矩阵 $\boldsymbol{\varepsilon}$ 为:

$$\boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_{11} & \varepsilon_{12} \\ \varepsilon_{21} & \varepsilon_{22} \end{pmatrix} = \begin{pmatrix} \varepsilon_{\mathrm{SS}} & \varepsilon_{\mathrm{SB}} \\ \varepsilon_{\mathrm{BS}} & \varepsilon_{\mathrm{BB}} \end{pmatrix} \tag{2.4.7}$$

其中, S 标记信号; B 标记本底; $\varepsilon_{\mathrm{SB}}$ 表示本底样本被分类器判为信号的判选率. 如若有一样本集, 其中模式类 $\omega_{\mathrm{S}}, \omega_{\mathrm{B}}$ 的样本数分别为 $n_{\mathrm{S}}, n_{\mathrm{B}}$, 被判选率矩阵为 $\boldsymbol{\varepsilon}$ 的分类器判别为模式类 $\omega_{\mathrm{S}}, \omega_{\mathrm{B}}$ 的样本数为 $\tilde{n}_{\mathrm{S}}, \tilde{n}_{\mathrm{B}}$. 则有

$$\tilde{\boldsymbol{n}} = \begin{pmatrix} \tilde{n}_{\mathrm{S}} \\ \tilde{n}_{\mathrm{B}} \end{pmatrix} = \boldsymbol{\varepsilon n} = \begin{pmatrix} \varepsilon_{\mathrm{SS}} & \varepsilon_{\mathrm{SB}} \\ \varepsilon_{\mathrm{BS}} & \varepsilon_{\mathrm{BB}} \end{pmatrix} \begin{pmatrix} n_{\mathrm{S}} \\ n_{\mathrm{B}} \end{pmatrix} = \begin{pmatrix} \varepsilon_{\mathrm{SS}} n_{\mathrm{S}} + \varepsilon_{\mathrm{SB}} n_{\mathrm{B}} \\ \varepsilon_{\mathrm{BS}} n_{\mathrm{S}} + \varepsilon_{\mathrm{BB}} n_{\mathrm{B}} \end{pmatrix}. \tag{2.4.8}$$

实际问题中往往更关心的是被分类器判为信号的样本数 $\tilde{n}_{\mathrm{S}}$, 希望其中包含的错判样本数 $\varepsilon_{\mathrm{SB}} n_{\mathrm{B}}$ 尽可能地少. 对于待分类的样本集, 本底样本数 n_{B} 是确定的, 因此只能要求 $\varepsilon_{\mathrm{SB}}$ 尽可能地小. 量 r 定义为分类器对信号样本的判选效率和本底样本被错判为信号样本的错分概率之比:

$$r = \frac{\varepsilon_{\mathrm{SS}}}{\varepsilon_{\mathrm{SB}}} \tag{2.4.9}$$

称为分类器的信号/本底分辨能力 (separation power). 分辨能力越大, 分类器判为信号的样本数中本底的污染越小. 分辨能力是分类器的一个非常重要的性能参数.

2.4.2 错误率的上界

从前面的讨论可知, 错误率的理论计算一般是相当困难的. 当不能从理论上直接计算出错误率时, 往往代之以寻找错误率的上界.

所谓的 Chernoff 上界被称为最小上界, 但它的计算比较复杂, 这里不加讨论. 有兴趣的读者可参阅参考文献 [5] 中相关部分的讨论. 利用 Bhattacharyya 系数确定错误率的上界相对地比较简单.

由 2.1 节的讨论已知, 对于两类问题, 样本 $\boldsymbol{x}$ 决策的条件错误率 $\varepsilon(e|\boldsymbol{x})$ 可表示为

$$\varepsilon(e|\boldsymbol{x}) = \min\left[q(\omega_1|\boldsymbol{x}), q(\omega_2|\boldsymbol{x})\right] \tag{2.4.10}$$

利用几何均值不等式 (即 $a > b > 0$ 时, $\sqrt{ab} > b$) 可得

$$\varepsilon(e|\boldsymbol{x}) \leqslant \sqrt{q(\omega_1|\boldsymbol{x})q(\omega_2|\boldsymbol{x})} \tag{2.4.11}$$

对条件错误率求期望值就得到错误率, 故有

$$\begin{aligned}\varepsilon(e) &= \int \varepsilon(e|\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\ &\leqslant \int \sqrt{q(\omega_1|\boldsymbol{x})\cdot q(\omega_2|\boldsymbol{x})}p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\ &= \sqrt{\pi(\omega_1)\pi(\omega_2)}\int \sqrt{p(\boldsymbol{x}|\omega_1)\cdot p(\boldsymbol{x}|\omega_2)}p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\ &= \sqrt{\pi(\omega_1)\pi(\omega_2)}\cdot \exp\left\{-\left[-\ln\int \sqrt{p(\boldsymbol{x}|\omega_1)\cdot p(\boldsymbol{x}|\omega_2)}p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right]\right\}\end{aligned}$$

定义 Bhattacharyya 系数 J_{B} 为

$$J_{\mathrm{B}} = -\ln\int \sqrt{p(\boldsymbol{x}|\omega_1)\cdot p(\boldsymbol{x}|\omega_2)}p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{2.4.12}$$

则错误率可表示为

$$\varepsilon(e) \leqslant \sqrt{\pi(\omega_1)\pi(\omega_2)}\cdot \exp(-J_{\mathrm{B}}) \tag{2.4.13}$$

上式右边即为利用 Bhattacharyya 系数确定错误率的上界. 计算该上界需要用到先验概率 $\pi(\omega_i)$ 和类条件概率密度 $p(\boldsymbol{x}|\omega_i)$ 的知识.

如果两类的类条件概率密度都服从正态分布, 即 $p(\boldsymbol{x}|\omega_i) \sim N(\boldsymbol{\mu}_i, \boldsymbol{V}_i),\ \ i = 1, 2$, 则可算出系数 J_{B} 为

$$J_{\mathrm{B}} = \frac{1}{8}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^{\mathrm{T}}\left(\frac{\boldsymbol{V}_1 + \boldsymbol{V}_2}{2}\right)^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) + \frac{1}{2}\ln\frac{|(\boldsymbol{V}_1 + \boldsymbol{V}_2)/2|}{\sqrt{|\boldsymbol{V}_1|\cdot|\boldsymbol{V}_2|}}. \tag{2.4.14}$$

2.4.3 利用检验样本集估计判选率矩阵和错误率

由上面的讨论可见, 即使各类别 ω_i 出现的先验概率 $\pi(\omega_i)$ 和样本 $\boldsymbol{x} \in \omega_i$ 时的条件概率密度 $p(\boldsymbol{x}|\omega_i)$ 都为已知, 在高维的情形下, 判选率矩阵和错误率在计算上也是相当复杂的, 即使错误率上界的计算也是相当复杂的, 有时在实际上无法进行.

由于判选率矩阵在模式识别中的重要性及计算上的复杂性, 促使人们研究对于判选率矩阵特别是错误率直接利用样本进行计算或估计的方法.

假定一个分类器用于对样本进行 c 个模式类 $\omega_1, \cdots, \omega_c$ 的判别. 可以按照如下方法得到分类器判选率矩阵 $\boldsymbol{\varepsilon}$ 的估计. 给定一检验样本集有 N 个已知类别的样本, 其中 $\omega_i(i = 1, 2, \cdots, c)$ 类的样本有 N_i 个, 显然 $N = \sum\limits_{i=1}^{c} N_i$.

假定分类器将一个 ω_j 类的样本判别为一个 ω_i 类的效率用 ε_{ij} 表示. 当把 ω_j 类的 N_j 个样本输入分类器后, 被判别为类别 ω_i 的样本数表示为 N_{ij}. N_{ij} 是一个随机变量, 其概率分布服从二项分布:

$$P(N_{ij}) = C_{N_j}^{N_{ij}}\varepsilon_{ij}^{N_{ij}}(1 - \varepsilon_{ij})^{N_j - N_{ij}}$$

ε_{ij} 的极大似然估计 $\hat{\varepsilon}_{ij}$ 为如下似然方程的解:

$$\frac{\partial \ln P(N_{ij})}{\partial \varepsilon_{ij}} = \frac{N_{ij}}{\varepsilon_{ij}} - \frac{N_j - N_{ij}}{1-\varepsilon_{ij}} = 0 \quad .$$

由此求得 ε_{ij} 的极大似然估计 $\hat{\varepsilon}_{ij}$:

$$\hat{\varepsilon}_{ij} = N_{ij}/N_j, \qquad i,j = 1,2,\cdots,c, \tag{2.4.15}$$

二项分布随机变量 N_{ij} 的期望值和方差为

$$\begin{aligned} E(N_{ij}) &= N_j \varepsilon_{ij} \\ V(N_{ij}) &= N_j \varepsilon_{ij}(1-\varepsilon_{ij}). \end{aligned}$$

因此估计量 $\hat{\varepsilon}_{ij}$ 的期望值和方差为:

$$E(\hat{\varepsilon}_{ij}) = E\left(\frac{N_{ij}}{N_j}\right) = \frac{E(N_{ij})}{N_j} = \varepsilon_{ij}, \tag{2.4.16}$$

$$V(\hat{\varepsilon}_{ij}) = V\left(\frac{N_{ij}}{N_j}\right) = \frac{V(N_{ij})}{N_j^2} = \frac{\varepsilon_{ij}(1-\varepsilon_{ij})}{N_j}. \tag{2.4.17}$$

由式 (2.4.16) 知 $\hat{\varepsilon}_{ij}$ 是 ε_{ij} 的无偏估计. 当 N_{ij} 充分大 (因而 N_j 必定充分大), 用式 (2.4.15) 和式 (2.4.17) 估计 ε_{ij} 及其方差是它们的真值的好的近似, 式中的 ε_{ij} 用 $\hat{\varepsilon}_{ij}$ 估计. 由式 (2.4.17) 可知, $\hat{\varepsilon}_{ij}$ 的标准偏差随着 $\sqrt{N_j}$ 的增大而减小.

我们还可以讨论一定置信水平下的置信区间 $(\varepsilon_1, \varepsilon_2)$ 与 $\hat{\varepsilon}_{ij}$ 和 N_j 的关系. 置信水平 CL 定义为 $\hat{\varepsilon}_{ij}$ 落在置信区间 $(\varepsilon_1, \varepsilon_2)$ 内的概率:

$$P(\varepsilon_1 \leqslant \hat{\varepsilon}_{ij} \leqslant \varepsilon_2) = CL. \tag{2.4.18}$$

图 2.6 是置信水平 $CL = 95\%$ 下的置信区间 $(\varepsilon_1, \varepsilon_2)$ 与 $\hat{\varepsilon}_{ij}$ 和 N_j 的关系曲线, 显然, 训练样本数 N_j 越大, $\hat{\varepsilon}_{ij}$ 的置信区间 $(\varepsilon_1, \varepsilon_2)$ 越小, 即 $\hat{\varepsilon}_{ij}$ 与真值 ε_{ij} 的差别越小. 例如当 $N_j = 50$ 而 $N_{ij} = 0$, 则 $\hat{\varepsilon}_{ij} = 0$. 从图 2.6 可知置信水平 $CL = 95\%$ 以下的置信区间 $(\varepsilon_1, \varepsilon_2)$ 为 $(0, 0.08)$, 即 ε_{ij} 在 $(0, 0.08)$ 范围内. 若 $N_j = 250$ 而 $N_{ij} = 0$, 则 $\hat{\varepsilon}_{ij} = 0$, ε_{ij} 在 $(0, 0.02)$ 范围内.

按照上述随机抽样方法得到判选率矩阵 ε 的估计后, 容易得到最小错误率贝叶斯决策规则分类器正确分类的效率的估计 $\hat{\varepsilon}(c)$:

$$\hat{\varepsilon}(c) = \sum_{i=1}^{c} \pi(\omega_i)\hat{\varepsilon}_{ii} = \sum_{i=1}^{c} \pi(\omega_i)\frac{N_{ii}}{N_i} \tag{2.4.19}$$

错分率的估计 $\hat{\varepsilon}(e)$

$$\hat{\varepsilon}(e) = 1 - \hat{\varepsilon}(c), \tag{2.4.20}$$

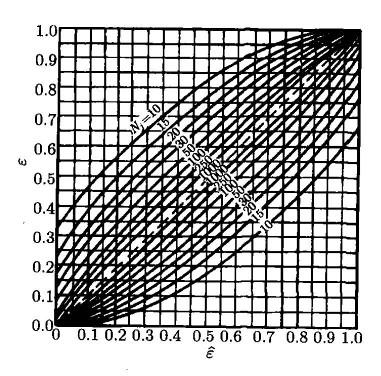


图 2.6 95%置信水平下的置信区间 $(\varepsilon_1, \varepsilon_2)$ 与 $\hat{\varepsilon}_{ij}$ 和 N_j 的关系曲线

及其方差的估计

$$V[\hat{\varepsilon}(c)] = V[\hat{\varepsilon}(e)] = V\left[\sum_{i=1}^{c}\pi(\omega_i)\hat{\varepsilon}_{ii}\right]$$

$$= \sum_{i=1}^{c}\frac{[\pi(\omega_i)]^2}{N_i}\varepsilon_{ii}(1-\varepsilon_{ii}) = \sum_{i=1}^{c}[\pi(\omega_i)]^2\frac{N_{ii}}{N_i^2}\left(1-\frac{N_{ii}}{N_i}\right). \quad (2.4.21)$$

如果检验样本集中不同类别的样本数 $N_i(i=1,2,\cdots,c)$ 是按照先验概率分配的, 即

$$\pi(\omega_i) = \frac{N_i}{N}, \qquad i=1,2,\cdots,c \quad (2.4.22)$$

代入式 (2.4.19) 和式 (2.4.21) 则有

$$\hat{\varepsilon}(c) = \sum_{i=1}^{c}\pi(\omega_i)\hat{\varepsilon}_{ii} = \frac{1}{N}\sum_{i=1}^{c}N_{ii}, \quad (2.4.23)$$

$$V[\hat{\varepsilon}(c)] = V[\hat{\varepsilon}(e)] = V\left[\sum_{i=1}^{c}\pi(\omega_i)\hat{\varepsilon}_{ii}\right] = \frac{1}{N^2}\sum_{i=1}^{c}N_{ii}\left(1-\frac{N_{ii}}{N_i}\right). \quad (2.4.24)$$

2.4.4 训练样本集和检验样本集的划分

如第一章中提到的, 为了要设计分类器, 通常要有类别已知的事例样本集. 当利用样本集来确定分类器的错误率, 同样要用到类别已知的事例样本集. 因此, 许多情况下样本集既用于分类器的设计 (或训练), 又用于确定分类器的错误率 (即其性能的检验). 类别已知的事例样本集在高能物理实验中通常有两种途径获得: 蒙特卡罗模拟数据和真实实验数据. 模拟数据原则上可以产生无限多的事例, 但受到

计算机机时的限制; 对于非常复杂的粒子反应过程, 模拟一个反应事例的计算机机时并不短, 因此模拟事例样本数实际上也是有限的. 至于真实的实验数据样本, 更是受到实验数据收集时间和反应截面的限制, 数据样本量往往不大. 因此怎样利用有限的类别已知的事例样本集来设计分类器, 并正确地估计它的错误率, 就是一个值得研究的课题.

在本节的讨论中, 假定用于分类器的训练和性能检验的样本集有 N 个样本, 其中 $\omega_i(i=1,2,\cdots,c)$ 类的样本有 N_i 个, 显然 $N=\sum_{i=1}^{c} N_i$.

有三种途径利用有限的样本进行分类器的训练和性能检验测试.

(1) 样本划分法

假定有 $\omega_i(i=1,2,\cdots,c)$ 类的样本 N_i 个, 它们被分为两组, 第一组称为设计集 (或训练集)N_i^D, 另一组称为检验集 (或测试集)N_i^{T}, 并有 $N_i=N_i^D+N_i^{\mathrm{T}}$. 其中样本集 N_i^D 用于分类器的设计, 样本集 N_i^{T} 用于分类器的性能检验. 显然, 要能训练出性能好的分类器, 并能估计出正确的错误率, N_i^D 和 N_i^{T} 都必须充分大. 因此本法仅适用于 N 充分大的情形.

(2) 留一法

本法适用于 N 比较小的情形. 这种方法中, 先选择样本 1 用作性能检验, 其余的 $N-1$ 个样本用作分类器的训练设计. 若样本 1 属于 ω_j, 而分类器对该样本的分类为 i, 则对变量 N_{ij} (初值为 0) 的值加 1. 然后选择样本 2 用作性能测试, 其余的 $N-1$ 个样本用作分类器的训练设计, 重复以上的步骤. 以此类推, 直到将所有 N 个样本完成同样的步骤为止. 这时, 变量 $N_{ij}(i,j=1,2,\cdots,c)$ 的值表示 N 个样本中 N_j 个 $\omega_j(j=1,2,\cdots,c)$ 类样本被分类器判为 $\omega_i(i=1,2,\cdots,c)$ 类样本的数目, 故判选率矩阵的矩阵元与式 (2.4.15) 式相同

$$\hat{\varepsilon}_{ij}=\frac{N_{ij}}{N_j}, \qquad i,j=1,2,\cdots,c.$$

估计量 $\hat{\varepsilon}_{ij}$ 的方差、分类器正确分类的效率 $\hat{\varepsilon}(c)$、错分率 $\hat{\varepsilon}(e)$ 及其方差依然可用式 (2.4.17) 和式 (2.4.19)~(2.4.24) 表示. 可以看出, 这种方法充分利用了仅有的 N 个样本, 一定程度上解决了样本划分法在 N 较小时的矛盾; 但是, 因为要进行 N 次分类器的训练, 计算量比较大.

(3) 分组轮换法

这是介于样本划分法和留一法之间的一种方法. 把 N 个样本分成 m 组, 每组含 N/m 个样本 (应为正整数). 首先抽第一组样本用作检验, 其余 $m-1$ 组样本用来训练分类器. 该分类器对第一组的样本逐一作分类, 若样本的类别为 j, 而分类器将其判别为 i 类样本, 则对变量 N_{ij} (初值为 0) 的值加 1. 对所有 m 组样本重复同样的步骤. 这时, 变量 $N_{ij}(i,j=1,2,\cdots,c)$ 的值表示 N 个样本中 N_j 个

$\omega_j(j=1,2,\cdots,c)$ 类样本被分类器判为 $\omega_i(i=1,2,\cdots,c)$ 类样本的数目. 于是式 (2.4.15), 式 (2.4.17) 和式 (2.4.19)~(2.4.24) 式的结果仍然适用. 这种方法在已知样本数 N 一定时, 对错误率的估计偏差小于样本划分法, 而计算量小于留一法 (只需作 m 次分类器训练).

2.4.5 利用判选率矩阵估计各类 "真实" 样本数

如若有一待分类的样本集, 其中模式类 $\omega_1,\omega_2,\cdots,\omega_c$ 的样本数分别为 n_1, $n_2,\cdots,n_c$, 被判选率矩阵为 $\boldsymbol{\varepsilon}$ 的分类器判别为模式类 $\omega_1,\omega_2,\cdots,\omega_c$ 的样本数为 $\tilde{n}_1,\tilde{n}_2,\cdots,\tilde{n}_c$. 如果定义 c 维向量

$$\boldsymbol{n}=(n_1,\quad n_2,\quad \cdots,\quad n_c)^{\mathrm{T}} \tag{2.4.25}$$

$$\tilde{\boldsymbol{n}}=(\tilde{n}_1,\quad \tilde{n}_2,\quad \cdots,\quad \tilde{n}_c)^{\mathrm{T}} \tag{2.4.26}$$

则有

$$\tilde{\boldsymbol{n}}=\boldsymbol{\varepsilon}\boldsymbol{n}. \tag{2.4.27}$$

反过来, 假定已知一个分类器的判选率矩阵为 $\boldsymbol{\varepsilon}$, 待分类的一个样本集中属于模式类 $\omega_1,\omega_2,\cdots,\omega_c$ 的样本数为未知 (用 $n_1,n_2,\cdots,n_c$ 表示), 但该分类器的输出值已知为 $\tilde{n}_1,\tilde{n}_2,\cdots,\tilde{n}_c$, 当判选率矩阵 $\boldsymbol{\varepsilon}$ 的逆矩阵 $\boldsymbol{\varepsilon}^{-1}$ 存在, 即其行列式不为 0: $\det\boldsymbol{\varepsilon}\neq 0$, 那么待分类样本集中各类的 "真实" 样本数 $n_1,n_2,\cdots,n_c$ 可用下式求得:

$$\boldsymbol{n}=\boldsymbol{\varepsilon}^{-1}\tilde{\boldsymbol{n}}. \tag{2.4.28}$$

或写成显著表式

$$n_i=\sum_{j=1}^{c}\varepsilon_{ij}^{-1}\tilde{n}_j,\qquad i=1,2,\cdots,c. \tag{2.4.29}$$

式中, ε_{ij}^{-1} 是判选率矩阵 $\boldsymbol{\varepsilon}$ 的逆矩阵 $\boldsymbol{\varepsilon}^{-1}$ 的元素. 该式告诉我们, 即使分类器对样本的种类存在误判, 只要它的判选率矩阵能够以足够好的精度加以确定, 那么, 从它判定样本集的结果 $\tilde{\boldsymbol{n}}$ 能够将样本集的原貌 $\boldsymbol{n}$ 以一定的精度 "复原" 回来.

当利用式 (2.4.29) 从分类器的判选率矩阵 $\boldsymbol{\varepsilon}$ 和分类器的输出值 $\tilde{n}_1,\tilde{n}_2,\cdots,\tilde{n}_c$ 计算其真实值 $n_1,n_2,\cdots,n_c$ 时, 需要注意如下的限制条件: 待分类的样本集中的样本种类 (用集合 $\boldsymbol{v}=\{v_1,v_2,\cdots,v_b\}$ 表示) 必须包含在确定分类器判选率矩阵时所用到检验样本集的模式类集合 $\boldsymbol{\omega}=\{\omega_1,\omega_2,\cdots,\omega_c\}$ 之中, 即必须有

$$\boldsymbol{v}\in\boldsymbol{\omega} \tag{2.4.30}$$

原因可以这样来阐明: 假定待分类的样本集中的样本种类比检验样本集模式类 $\omega_1,\omega_2,\cdots,\omega_c$ 多出一种, 用 ω_{c+1} 表示. 当用分类器来判别待分类的样本集中的

样本种类时, 它对样本类别的输出值只可能为 $\omega_1, \omega_2, \cdots, \omega_c$, 不可能为 ω_{c+1}, 于是该分类器会将待分类的样本集中的 ω_{c+1} 类样本以某种概率判定为 $\omega_1, \omega_2, \cdots, \omega_c$ 样本之一. 这与分类器效率 ε_{ij} 等于分类器将一个 ω_j 类的样本判别为一个 ω_i 类 $(i, j = 1, 2, \cdots, c)$ 的定义不符, 于是 $\hat{\varepsilon}_{ij}$ 的表式 (2.4.15) 不再适用, 2.4.3 节的其他公式也不再适用. 通常训练分类器的训练样本集的模式类与检验分类器的检验样本集的模式类数目和种类是相同的, 因此结论是待分类的样本集中的样本种类的集合 $\boldsymbol{v}$ 必须包含在训练分类器时所用到的模式类集合 $\boldsymbol{\omega}$ 之中. 这一结论在分类器的实际使用中十分重要. 例如我们对于一个粒子反应过程的研究中要鉴别带电粒子 $\mathrm{e}^{\pm}, \mu^{\pm}$, 但实际的数据样本中包含了带电粒子 $\mathrm{e}^{\pm}, \mu^{\pm}, \pi^{\pm}, \mathrm{K}^{\pm}, \mathrm{p}, \bar{\mathrm{p}}$, 为了得到正确的判选率矩阵 $\boldsymbol{\varepsilon}$, 训练粒子鉴别的分类器和用样本确定 $\boldsymbol{\varepsilon}$ 时, 训练样本集和检验样本集必须包含全部带电粒子 $\mathrm{e}^{\pm}, \mu^{\pm}, \pi^{\pm}, \mathrm{K}^{\pm}, \mathrm{p}, \bar{\mathrm{p}}$ 的样本, 而不能仅包含 $\mathrm{e}^{\pm}, \mu^{\pm}$ 样本.

下面来讨论用式 (2.4.29) "复原" 回来的样本集中 ω_i 类的 "真实" 样本数 $n_i (i = 1, 2, \cdots, c)$ 的误差. 将 "间接测定量" $\boldsymbol{n}$ 视为 "直接测定量" $\tilde{\boldsymbol{n}}$ 的函数, 利用误差传播公式可得

$$V_{kl}(\boldsymbol{n}) \cong \sum_{i=1}^{c} \sum_{j=1}^{c} \left(\frac{\partial n_k}{\partial \tilde{n}_i} \cdot \frac{\partial n_l}{\partial \tilde{n}_j} \right)_{\tilde{\boldsymbol{n}} = \hat{\tilde{\boldsymbol{n}}}} V_{ij}(\tilde{\boldsymbol{n}}), \qquad k, l = 1, 2, \cdots, c \tag{2.4.31}$$

由式 (2.4.29) 可知

$$\frac{\partial n_k}{\partial \tilde{n}_i} = \frac{\partial}{\partial \tilde{n}_i} \sum_{j=1}^{c} \varepsilon_{kj}^{-1} \tilde{n}_j = \varepsilon_{ki}^{-1}, \qquad k = 1, 2, \cdots, c.$$

代入式 (2.4.31) 得

$$\begin{aligned} V(n_k) = V_{kk}(\boldsymbol{n}) &\cong \sum_{i=1}^{c} \sum_{j=1}^{c} \varepsilon_{ki}^{-1} \varepsilon_{kj}^{-1} V_{ij}(\tilde{\boldsymbol{n}}) \\ &= \sum_{i=1}^{c} \sum_{j=1}^{c} \varepsilon_{ki}^{-1} \varepsilon_{kj}^{-1} \rho_{ij}(\tilde{\boldsymbol{n}}) \sigma_i \sigma_j, \quad k = 1, 2, \cdots, c \end{aligned} \tag{2.4.32}$$

式中, $\rho_{ij}(\tilde{\boldsymbol{n}})$ 是 $\tilde{n}_i$ 和 $\tilde{n}_j$ 之间的相关系数, $\tilde{\sigma}_i$ 是 $\tilde{n}_i$ 的标准偏差.

可将随机变量 $\tilde{\boldsymbol{n}}$ 考虑为 c 维的多项分布:

$$M(\tilde{\boldsymbol{n}}; \tilde{n}, \boldsymbol{p}) = \frac{\tilde{n}!}{\tilde{n}_1! \tilde{n}_2! \cdots \tilde{n}_c!} p_1^{\tilde{n}_1} p_2^{\tilde{n}_2} \cdots p_c^{\tilde{n}_c}. \tag{2.4.33}$$

其中参数 $\tilde{n}$ 为 n 个待分类样本被分类器判定为 $i (i = 1, 2, \cdots, c)$ 类样本数的总和:

$$\tilde{n} \equiv \sum_{i=1}^{c} \tilde{n}_i, \tag{2.4.34}$$

$p_i(i=1,2,\cdots,c)$ 表示分类器判定一个事例为一个 i 类事例的概率.

当待分类样本集中的模式类集合 $\boldsymbol{v}$ 满足式 (2.4.30) 时, 分类器将待分类样本集中任一模式类的样本总是判别为类别 $\omega_1,\omega_2,\cdots,\omega_c$ 的样本之一, 这时有

$$\tilde{n}\equiv\sum_{i=1}^{c}\tilde{n}_i=\sum_{i=1}^{c}n_i\equiv n, \tag{2.4.35}$$

即参数 $\tilde{n}$ 为待分类样本集的样本总数 n, 为一个常数. 这时, 多项分布有如下性质:

均值 $$E\left(\tilde{n}_j\right)=\tilde{n}p_j\ ,\ \ j=1,2,\cdots,c, \tag{2.4.36}$$

方差 $$V\left(\tilde{n}_j\right)=\tilde{n}p_j\left(1-p_j\right)\ ,\quad j=1,2,\cdots,c, \tag{2.4.37}$$

协方差 $$V_{ij}=Cov(\tilde{n}_i,\tilde{n}_j)=-\tilde{n}p_ip_j,\ i\neq j,\ i,j=1,\cdots,l. \tag{2.4.38}$$

用 $\hat{p}_j=\tilde{n}_j/\tilde{n}$ 作为 p_j 的估计, 并将式 (2.4.37~38) 代入式 (2.4.32), 即有

$$\begin{aligned}V(n_k)&=\sum_{i=1}^{c}\left(\varepsilon_{ki}^{-1}\right)^2\tilde{n}p_i(1-p_i)-\sum_{i=1}^{c}\sum_{j=1,j\neq i}^{c}\varepsilon_{ki}^{-1}\varepsilon_{kj}^{-1}\tilde{n}p_ip_j\\&=\sum_{i=1}^{c}\left(\varepsilon_{ki}^{-1}\right)^2\tilde{n}_i\left(1-\frac{\tilde{n}_i}{\tilde{n}}\right)-\sum_{i=1}^{c}\sum_{j=1,j\neq i}^{c}\varepsilon_{ki}^{-1}\varepsilon_{kj}^{-1}\frac{\tilde{n}_i\tilde{n}_j}{\tilde{n}}.\end{aligned}\qquad k=1,2,\cdots,c. \tag{2.4.39}$$

2.4.6 分类器判定的“信号”样本中错判事例的扣除

许多实际问题中涉及的往往是两类样本的分类问题, 即信号样本和本底样本的判别. 如 2.4.1 小节所述, 被判选率矩阵为 $\boldsymbol{\varepsilon}$ 的分类器判别为模式类 $\omega_{\rm S}$ 的样本 (即“信号”事例) 数 $\tilde{n}_{\rm S}=\varepsilon_{\rm SS}n_{\rm S}+\varepsilon_{\rm SB}n_{\rm B}$ 中, 除了被正确地判别的真实信号事例数 $\varepsilon_{\rm SS}n_{\rm S}$, 还包含了错判样本数 $\varepsilon_{\rm SB}n_{\rm B}$. 在某些情形下, 后者甚至大于前者. 因此就提出了将错判样本数从分类器判定的“信号”事例中扣除的要求.

这种情况在粒子物理实验数据分析中具有典型意义. 例如为了测量 $\psi(2S)\to\pi^0 J/\psi$ 的衰变分支比, 我们利用北京正负电子对撞机在质心系能量 3.686GeV 处产生 $\psi(2S)$ 粒子: $e^+e^-\to\psi(2S)$. 由于 $\psi(2S)$ 粒子有很多衰变道, 我们需要从中把信号事例 $\psi(2S)\to\pi^0 J/\psi$ 挑选出来, 这一过程在粒子物理实验数据分析中称为 (信号) 事例选择, 相应于利用一个事例分类器将信号事例从全部事例中判别出来.

在这项具体研究中, 问题的复杂性还在于 π^0 和 J/ψ 都是寿命极短的粒子, 它们立即衰变: $\pi^0\to\gamma\gamma$, $J/\psi\to e^+e^-,\mu^+\mu^-$, 因此探测器对于信号事例能探测的末态是 $\gamma\gamma e^+e^-,\gamma\gamma\mu^+\mu^-$. 按照粒子物理理论, 由 J/ψ 衰变产生的 $e^+e^-,\mu^+\mu^-$ 的不变质量应当在 J/ψ 质量附近, 我们可以按照 $\gamma\gamma e^+e^-,\gamma\gamma\mu^+\mu^-$ 末态事例的特征以及 $e^+e^-,\mu^+\mu^-$ 的不变质量应当在 J/ψ 质量附近这一特点设计事例选择程序 (即事例

分类器), 将 $\psi(2S) \to \gamma\gamma J/\psi(\to e^+e^-)$ 和 $\psi(2S) \to \gamma\gamma J/\psi(\to \mu^+\mu^-)$ 的候选事例 (即分类器判定的 "信号" 事例) 判选出来. 在这一分类器中两光子不变质量 $M_{\gamma\gamma}$ 是用来进行事例判选的特征变量之一. 分类器判定的 "信号" 事例的 $M_{\gamma\gamma}$ 分布如图 2.7 所示[16]. 由图可见, 在 0.135GeV 附近出现明显的峰状结构, 这是由于 π^0 衰变产生的两个光子的不变质量应当在 π^0 的质量即 0.135GeV 附近, 而 $\psi(2S) \to \pi^0 J/\psi$ 信号以外的本底事例的 $M_{\gamma\gamma}$ 分布则呈现比较平坦的分布, 如图中的两条平滑曲线所示. 将实验测量值用代表信号事例的峰状曲线和代表本底事例的平坦曲线作拟合, 峰状曲线的面积即是真正的 $\psi(2S) \to \pi^0 J/\psi$ 信号事例数, 即实现了从分类器判定的 "信号" 事例数中扣除本底事例的污染. 代表信号事例的峰状曲线和代表本底事例的平坦曲线的函数形式应当根据对于信号事例和本底事例的物理过程的理解来确定, 在目前的例子中它们分别是高斯函数和多项式函数.

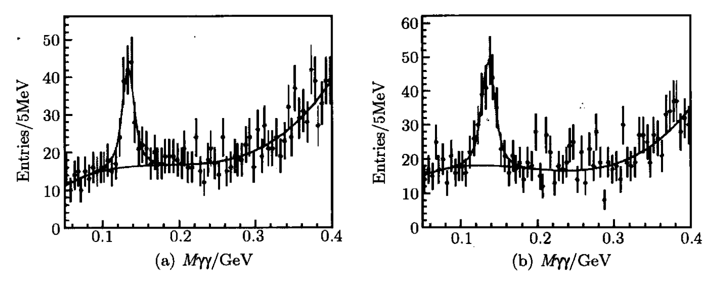


图 2.7 事例的两光子不变质量谱

图中数据点为实验测量值, 峰状曲线是信号事例的拟合曲线, 平坦曲线是本底事例的拟合曲线

(a) $\psi(2S) \to \gamma\gamma J/\psi(\to e^+e^-)$ (b) $\psi(2S) \to \gamma\gamma J/\psi(\to \mu^+\mu^-)$

应当指出, 分类器对于 "信号" 事例和 "本底" 事例的区分完全依赖于单个事例的特征向量的数值, 而上述从分类器判定的 "信号" 事例数中扣除本底事例的污染的方法则依靠某一特征变量的整体分布, 这在分类器中是无法完成的; 只有在分类器完成事例分类后, 对于其中的某个特征变量值的分布 (信号和本底事例的该特征变量的分布有明显的不同) 进行拟合, 才能将真正的信号事例的贡献分离出来. 例如在图 2.7 中, 我们即使在分类器中把特征变量 $M_{\gamma\gamma}$ 落入 [0.1, 0.2]GeV 的事例才选为 "信号" 事例, 仍然有相当多的本底事例会被判为 "信号" 事例 (即 $M_{\gamma\gamma}$ 落入 [0.1, 0.2]GeV 的平滑本底曲线下的部分事例). 为了能够对本底曲线进行拟合, 在设计分类器时, 该特征变量的区间应当选得比真正的信号事例区要宽一些, 这样才能根据信号事例区两边的本底区间里 (即所谓的边带区) 的该特征变量的分布来拟合本底曲线的形状.

2.5 讨　论

本章讨论了基于最小错误率的贝叶斯决策，以及对于两类问题要求其中的一类错误率不得大于某个给定常数而使另一类错误率尽可能地小的 Neyman-Pearson 决策方法. 贝叶斯决策的另一种重要方法是基于最小风险的贝叶斯决策，其基本思想是采用每一个决策时，都使其条件风险最小，则对所有的 x 作出决策时，其期望风险也必然最小. 此外，还有所谓的最小最大决策，其基本思想是如何使最大可能的风险达到最小. 这里对这两种决策方法没有加以讨论，有兴趣的读者可阅读相关的文献[5,6]. 以贝叶斯决策为核心内容的统计决策理论是统计模式识别的重要基础，依据它设计的分类器具有理论上的最优性能，即它的分类错误率或风险在所有可能的分类器中是最小的，因此经常用来作为衡量其他分类器设计方法优劣的标准.

既然已经有了最优的分类器，为什么还有必要研究其他方法呢? 这是由于贝叶斯决策分类有两个重要的前提，即本节一开始提到的：(1) 要决策分类的类别数 c 是已知的，(2) 要求对应于各类别 ω_i 出现的先验概率 $\pi(\omega_i)$ 和样本 $\boldsymbol{x} \in \omega_i$ 时的条件概率密度 $p(\boldsymbol{x}|\omega_i)$ 都是已知的. 要求类别数已知在实际的监督模式识别问题中毫不困难，因为这是我们分类的目标. 问题的困难之处在于第二个条件在实际问题中通常是不满足的. 因此必须寻找先验概率 $\pi(\omega_i)$ 和类条件概率密度 $p(\boldsymbol{x}|\omega_i)$ 未知情形下的分类方法.

为了设计这种条件下的分类方法，首先可以想到的途径是设法估计出先验概率和类条件概率密度. 前者可以根据实验数据中各类事例样本比例的先验知识得到，而后者的估计却需要统计学的一套复杂的方法. 因而实际问题中用贝叶斯决策理论设计分类器，其关键在于如何进行类条件概率密度的估计. 第七章中的概率密度估计量方法，讨论了利用样本数据构造类条件概率密度的估计量，然后再用贝叶斯方法对未知样本进行分类.

能否不按照上述思路而直接依靠训练样本设计分类器呢? 事实上，分类器就是确定一个 (或一系列) 判别函数 (或决策面)，如果从要解决的问题和训练样本出发直接求出判别函数，就可以不必进行概率密度的估计. 在某些情况下，判别函数具有较简单的形式，比如线性或二次函数的形式. 如果事先能够确定判别函数或决策面方程的形式 (或为了分类器设计的简便将判别函数设定为某种简单的形式)，再通过训练样本确定其中的参数，就能够简便地设计出分类器. 这就是从样本出发直接设计分类器的思路. 这类方法往往更具有实用价值. 本书第四和第五章中的方法与第七章中的大部分方法都属于这类方法.

上面提到的从样本出发直接设计分类器的思路都是分两步来解决模式识别问题的，即首先根据已知数据 (训练样本) 设计分类器，然后用它对待定样本进行分

类. 能否直接从训练样本出发对待定样本进行分类呢? 第六章中的近邻法就是采用了这种思路.

本章 2.4 节对分类器效率、错误率和判选率矩阵及相关的问题作了简略的讨论, 除了明确针对贝叶斯决策的部分之外, 其一般原则和论述同样适用于其他判别方法. 其中, 利用数据样本直接估计判选率矩阵的方法具有很大的实用价值.

第三章　线性判别方法

3.1　线性判别函数

3.1.1　线性判别函数的基本概念

假定有 c 个模式类, 用 $\omega_1,\cdots,\omega_c$ 表示. 所有的样本已经映射到特征空间里. 特征空间的维数用 n 表示, 每个样本就是 n 维特征空间的一个点. 在特征空间中, 属于一个模式类 ω_i 的点集与属于另一个模式类 ω_j 的点集总在某种程度上互相分离. 若能找到一个判别方法, 将不同类的点集分离开来, 就实现了不同模式类的判别.

对于最简单的两类问题, $c=2$. 两类问题是模式分类的基础, 多类问题可递归地用两类问题来解决. 假定特征维数 $n=2$, 样本点或特征向量可表示为

$$\boldsymbol{x}=(x_1,x_2)^{\mathrm{T}}.$$

假定已知两个模式类的样本点在特征空间中的分布如图 3.1 所示. 可以找到一个边界, 满足方程

$$g(\boldsymbol{x})=0, \tag{3.1.1}$$

它把特征空间划分成两个类型区域, 并且有

$$\begin{aligned} &g(\boldsymbol{x})>0, &&\text{则}\quad \boldsymbol{x}\in\omega_1,\\ &g(\boldsymbol{x})<0, &&\text{则}\quad \boldsymbol{x}\in\omega_2,\\ &g(\boldsymbol{x})=0, &&\text{则不可判别.}\end{aligned} \tag{3.1.2}$$

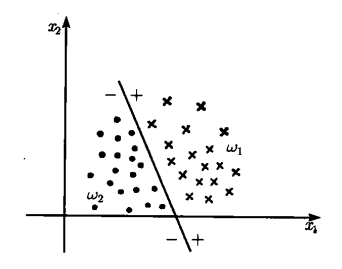


图 3.1　两类模式的判别

那么, 对于任意一个待定模式的新的样本点, 就可用式 (3.1.1)~(3.1.2) 来确定其模式属于 ω_1 或 ω_2. 函数 $g(\boldsymbol{x})$ 称为判别函数, 式 (3.1.2) 描述了判别规则, 而式 (3.1.1) 给定了区分界面. 一般地, 对于 n 维特征空间, $g(\boldsymbol{x})=0$ 称为决策面方程; 在三维特征空间的情形下, 它表示判别界面; 对于两维和一维特征空间, 它退化为分界线和分界点. 当 $n>3$, 判别边界为超表面.

根据判别函数 $g(\boldsymbol{x})$ 为特征向量 $\boldsymbol{x}$ 的一次 (线性) 函数或非线性函数, 称为线性判别函数或非线性判别函数. 以上论述虽然是从特征维数 $n=2$ 开始叙述的, 却对于任意 n 都适用.

两类情况下线性判别函数的一般表式

$$g(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+w_0, \tag{3.1.3}$$

式中, $\boldsymbol{x}$ 是 n 维特征向量; $\boldsymbol{w}$ 称为权向量:

$$\boldsymbol{x}=(x_1,\cdots,x_n)^{\mathrm{T}}, \qquad \boldsymbol{w}=(w_1,\cdots,w_n)^{\mathrm{T}}, \tag{3.1.4}$$

w_0 是个常数, 称为阈值权.

由于 $g(\boldsymbol{x})=0$ 为决策面方程, 且 $g(\boldsymbol{x})$ 为特征向量 $\boldsymbol{x}$ 的线性函数, 则对于三维特征空间的情形, 它表示决策平面; 对于两维和一维特征空间, 它退化为直线和分界点. 当 $n>3$, 判别边界为超平面. 这样, 可以给出两类问题线性可分性的定义如下: 当属于两个类型的样本在特征空间里能被一个超平面区分时, 它们是线性可分的.

下面, 我们来讨论超平面的一些性质.

假定特征向量 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$ 都在决策面 H 上, 则有

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_1+w_0=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_2+w_0, \tag{3.1.5}$$

或

$$\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{x}_1-\boldsymbol{x}_2)=0. \tag{3.1.6}$$

这时 $(\boldsymbol{x}_1-\boldsymbol{x}_2)$ 是决策面 H 上的任意一个向量, 所以式 (3.1.6) 表明权向量 $\boldsymbol{w}$ 与超平面 H 上的任意向量垂直, 即 $\boldsymbol{w}$ 是超平面 H 的法向量. 超平面 H 把特征空间分成两个半空间, 即 ω_1 模式类的决策域 R_1 和 ω_2 模式类的决策域 R_2. 因为按式 (3.1.2), 当 $\boldsymbol{x}$ 在 R_1 中时, $g(\boldsymbol{x})>0$, 所以决策面 H 的法向量 $\boldsymbol{w}$ 是指向 R_1 的. 因此, 有时称 R_1 中的所有 $\boldsymbol{x}$ 在超平面 H 的正侧, R_2 中的所有 $\boldsymbol{x}$ 在超平面 H 的负侧.

判别函数 $g(\boldsymbol{x})$ 可以看成是特征空间中某点 $\boldsymbol{x}$ 到超平面 H 的距离的一种代数度量, 如图 3.2. 若把 $\boldsymbol{x}$ 写为

$$\boldsymbol{x}=\boldsymbol{x}_{\mathrm{p}}+r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}, \tag{3.1.7}$$

式中, r 是 $\boldsymbol{x}$ 到 H 的垂直距离; $\boldsymbol{x}_{\rm p}$ 是 $\boldsymbol{x}$ 到 H 的垂直线与 H 的交点; $\boldsymbol{w}/\|\boldsymbol{w}\|$ 是法向量 $\boldsymbol{w}$ 方向的单位向量.

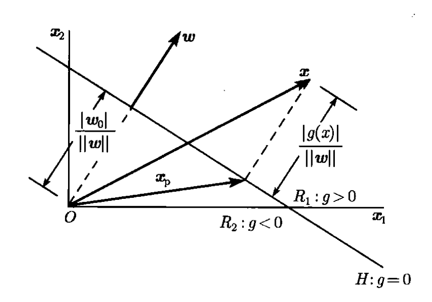


图 3.2 线性判别函数

将式 (3.1.7) 代入式 (3.1.3), 可得

$$g(\boldsymbol{x})=\boldsymbol{w}^{\rm T}\left(\boldsymbol{x}_{\rm p}+r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}\right)+w_0=\boldsymbol{w}^{\rm T}\boldsymbol{x}_{\rm p}+w_0+r\frac{\boldsymbol{w}^{\rm T}\boldsymbol{w}}{\|\boldsymbol{w}\|},$$

注意到 $\boldsymbol{x}_{\rm p}$ 是超平面 H 上的一个点, 故有 $\boldsymbol{w}^{\rm T}\boldsymbol{x}_{\rm p}+w_0=0$, 因此

$$g(\boldsymbol{x})=r\|\boldsymbol{w}\|,$$

或写为

$$r=\frac{g(\boldsymbol{x})}{\|\boldsymbol{w}\|}. \tag{3.1.8}$$

当 $\boldsymbol{x}$ 为原点, 由式 (3.1.3) 知

$$g(\boldsymbol{x}=0)=w_0. \tag{3.1.9}$$

将式 (3.1.9) 代入式 (3.1.8), 就得到原点到超平面 H 的距离

$$r_0=\frac{w_0}{\|\boldsymbol{w}\|}. \tag{3.1.10}$$

若 $w_0>0$, 则原点在超平面 H 的正侧; 若 $w_0<0$, 则原点在超平面 H 的负侧; 若 $w_0=0$, 则 $g(\boldsymbol{x})$ 具有齐次形式 $g(\boldsymbol{x})=\boldsymbol{w}^{\rm T}\boldsymbol{x}$, 说明超平面 H 通过原点.

总之, 利用线性判别函数进行决策, 就是用一个超平面把特征空间划分为两个模式类别区域. 超平面的方向由权向量 $\boldsymbol{w}$ 确定, 它的位置由阈值权 w_0 确定. 判别函数 $g(\boldsymbol{x})$ 正比于 $\boldsymbol{x}$ 点到超平面 H 的代数距离 (带正负号), 当 $\boldsymbol{x}$ 点在超平面 H 的正侧时, $g(\boldsymbol{x})>0$; 在负侧时, $g(\boldsymbol{x})<0$.

3.1.2　广义线性判别函数

考虑图 3.3 所示的两类问题, 设有一维样本空间 X, 判别函数 $g(x)$ 如图中曲线所示, 即 $x > a$ 或 $x < b$ 时, x 属于 ω_1 类; 如果 $b < x < a$, 则 x 属于 ω_2 类. 显然, 没有任何一个线性判别函数能够实现这样的判别问题. 这说明线性判别函数虽然简单, 但是有较大的局限性, 不适用于非凸决策区域和多连通区域的划分问题.

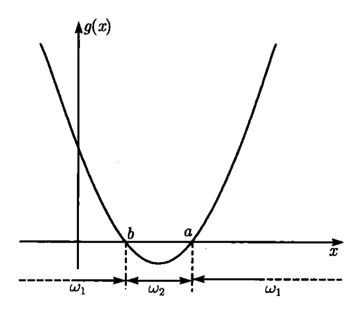


图 3.3　二次判别函数的例

但是, 如果建立一个二次判别函数

$$g(x) = (x-a)(x-b) \tag{3.1.11}$$

则可以很好地解决上述分类问题, 决策规则是

$$\begin{aligned} g(x) > 0, &\qquad \text{则决策 } x \in \omega_1, \\ g(x) < 0, &\qquad \text{则决策 } x \in \omega_2. \end{aligned} \tag{3.1.12}$$

二次判别函数可写成如下一般形式

$$g(x) = c_0 + c_1 x + c_2 x^2. \tag{3.1.13}$$

如果适当选择 $\boldsymbol{x} \to \boldsymbol{y}$ 的映射, 则可把 x 的二次判别函数化为 $\boldsymbol{y}$ 的线性函数

$$g(x) = \boldsymbol{v}^{\mathrm{T}} \boldsymbol{y} = \sum_{j=1}^{3} v_j y_j, \tag{3.1.14}$$

式中

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}, \qquad \boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix}.$$

$g(\boldsymbol{x}) = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}$ 称为广义线性判别函数, v 称为广义权向量.

一般说来, 对于任意高次的判别函数 $g(x)$, 都可以通过适当的变换, 化为广义线性判别函数来处理. $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}$ 不是 x 的线性函数, 但却是 $\boldsymbol{y}$ 的线性函数. $g(x) = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{y} = 0$ 在 $\boldsymbol{Y}$ 空间确定了一个通过原点的超平面. 这样就可以利用线性判别函数的简单性来解决较为复杂的非线性问题. 遗憾的是, 经过这种变换, 维数增加了, 这将使问题陷入所谓的 "维数灾难". 但若把式 (3.1.3) 定义的线性判别函数写成下面的形式:

$$g(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + w_0 = w_0 + \sum_{j=1}^{n} w_j x_j = \sum_{j=1}^{\hat{n}} v_j y_j = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}, \tag{3.1.15}$$

其中

$$\boldsymbol{y} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} 1 \\ \boldsymbol{x} \end{bmatrix}, \qquad \boldsymbol{v} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} w_0 \\ \boldsymbol{w} \end{bmatrix},$$

则它是广义线性判别函数的一个特例. 式 (3.1.15) 称为线性判别函数的齐次简化, $\boldsymbol{y} = (1, \boldsymbol{x})^{\mathrm{T}}$ 叫做增广样本向量, $\boldsymbol{v} = (w_0, \boldsymbol{w})^{\mathrm{T}}$ 叫做增广权向量, 它们是 $\hat{n} = n + 1$ 维向量. 虽然 $\boldsymbol{y}$ 比 $\boldsymbol{x}$ 增加了一维, 但保持了样本间的欧氏距离不变, 变换后的样本向量仍然全部位于 n 维子空间, 即原来的 $\boldsymbol{X}$ 空间中. 方程

$$\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y} = 0, \tag{3.1.16}$$

在 $\boldsymbol{Y}$ 空间中确定了一个通过原点的超平面 $\hat{H}$, 它对 n 维子空间的划分与原决策面

$$g(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + w_0 = 0,$$

对原 $\boldsymbol{X}$ 空间的划分完全相同. $\boldsymbol{Y}$ 空间中任意一点 $\boldsymbol{y}$ 到超平面 $\hat{H}$ 的距离可根据式 (3.1.8) 求得:

$$\hat{\boldsymbol{r}} = \frac{g(\boldsymbol{x})}{\|\boldsymbol{v}\|} = \frac{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}}{\|\boldsymbol{v}\|}. \tag{3.1.17}$$

现在, 我们可以对线性可分性的概念作如下的阐述: 假设已知一组容量 N 的样本集 $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N$, 其中 $\boldsymbol{y}_i$ 是 $\hat{n} = n + 1$ 维增广样本向量, 分别来自模式类 ω_1 和 ω_2. 如果存在一个线性分类器能把每个样本正确分类, 即如果存在一个权向量 $\boldsymbol{v}$, 使得对于任意的 $\boldsymbol{y} \in \omega_1$, 都有 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y} > 0$; 而对于任意的 $\boldsymbol{y} \in \omega_2$, 都有 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y} < 0$, 则称该样本集是线性可分的; 否则称为线性不可分的. 反过来, 如果样本集是线性可分的, 则必定存在一个权向量 $\boldsymbol{v}$, 能把该样本集的每个样本正确地分类.

3.1.3　线性分类器的设计

所谓线性分类器的设计, 就是利用模式类已知的训练样本集建立线性判别函数式 (3.1.3) 或广义线性判别函数式 (3.1.15). 这两个式子中只有权向量 $\boldsymbol{w}$ 和阈值权 w_0 或增广权向量 $\boldsymbol{v}$ 是未知的. 权向量 $\boldsymbol{w}$ 和阈值权 w_0 或增广权向量 $\boldsymbol{v}$ 的介不是唯一的, 而可以存在多个介. 设计线性分类器的过程, 实际上是寻找最优的 $\boldsymbol{w}$ 和 w_0 的过程. 权向量 $\boldsymbol{w}$ 和阈值权 w_0 或增广权向量 $\boldsymbol{v}$ 的介通常用准则函数 J 来寻找, 最优的 $\boldsymbol{w}$ 和 w_0 值通常出现在准则函数 J 的极值点上. 这样, 线性分类器的设计问题就转化为利用训练样本集寻找准则函数 J 的极值点 $\boldsymbol{w}^*$ 和 w_0^* 或 $\boldsymbol{v}^*$ 的问题了.

于是, 设计线性分类器的主要步骤可以概括如下:

(1) 事先要有一组具有类别标志 (即类别已知) 的训练样本集 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N\}$. 如有必要, 将训练样本集 $\boldsymbol{X}$ 转换成增广样本集 $\boldsymbol{Y}$.

(2) 根据实际情况确定一个准则函数 J, 它必须满足: (a) J 是样本集 $\boldsymbol{X}$ 和 $\boldsymbol{w},w_0$ 或 $\boldsymbol{v}$ 的函数. (b) J 的值反映分类器的性能, 它的极值解对应于 “最优” 的决策.

(3) 用最优化技术求出准则函数 J 的极值解 $\boldsymbol{w}^*$ 和 w_0^* 或 $\boldsymbol{v}^*$.

这样就可以得到线性判别函数 $g(\boldsymbol{x})=\boldsymbol{w}^{*\mathrm{T}}\boldsymbol{x}+w_0^*$ 或 $g(\boldsymbol{x})=\boldsymbol{v}^{*\mathrm{T}}\boldsymbol{y}$.

对于未知类别的样本 $\boldsymbol{x}_k$, 只要计算 $g(\boldsymbol{x}_k)$, 然后根据决策规则式 (3.1.2), 就可判断 $\boldsymbol{x}_k$ 所属的模式类别.

3.2　Fisher 线性判别

关于线性判别函数的分析, 历史上是从 R.A.Fisher 的经典论文 (1936 年) 开始的[17]. Fisher 方法涉及维数降低的问题. 因为低维空间会给问题的分析和计算带来很多方便, 而高维空间往往会使某些解析和计算方法难以实现, 即所谓 “维数灾难”, 所以在许多情况下, 降低维数就成为处理实际问题的关键之一.

为了实现降维, 可以考虑把 n 维特征空间的样本投影到一条直线上, 即把特征空间压缩成一维, 这在数学上容易实现. 但是, 即使样本在 n 维特征空间聚集为相互分离的点群, 它们在一条任意直线上的投影却可能相互混杂在一起而无法区分. 因此这根直线方向的选择非常重要. 一般情况下, 总可以找到某个方向, 使不同模式类的样本在该直线上的投影是最容易区分开的. 如何找到最好的直线方向, 如何实现在该方向上的投影变换, 就是 Fisher 方法要解决的基本问题 (见图 3.4).

下面讨论二类模式的 Fisher 线性判别方法.

假定我们处理的是 ω_1/ω_2 两类模式的分类问题, 并已有 N 个 n 维训练样本 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N\}$, 其中 N_1 个样本属于 ω_1 模式类记为子集 $\boldsymbol{X}_1$, N_2 个样本属

于 ω_2 类记为子集 $\boldsymbol{X}_2$.

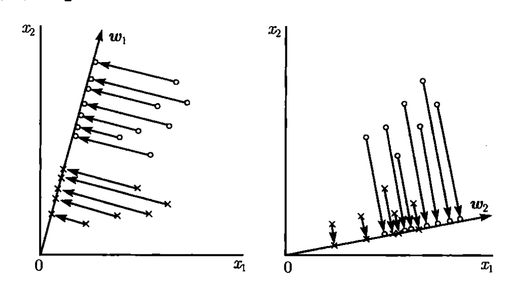


图 3.4 Fisher 线性判别的基本原理

图中圆圈和叉表示不同类的样本点, 由图可见, 直线 $\boldsymbol{w}_1$ 对于两类样本点的区分比直线 $\boldsymbol{w}_2$ 好

对 n 维向量 $\boldsymbol{x}_i, i=1,2,\cdots,N$ 作如下变换:

$$y_i = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i, \qquad i=1,2,\cdots,N \tag{3.2.1}$$

y_i 是 n 维向量 $\boldsymbol{x}_i$ 通过变换 $\boldsymbol{w}=(w_1,w_2,\cdots,w_n)^{\mathrm{T}}$ 得到的一维标量, 这就实现了从 n 维空间到一维空间的数学变换. y_i, $i=1,2,\cdots,N$ 是 n 维向量训练样本 $\boldsymbol{X}$ 的对应一维样本的集合 Y; 并可划分为对应于 $\boldsymbol{X}_1$ 和 $\boldsymbol{X}_2$ 的两个子集 Y_1 和 Y_2. n 维向量 $\boldsymbol{w}$ 定义了 n 维特征空间中的一条直线. 如果取 $\|\boldsymbol{w}\|=1$, 则 y_i 就是 $\boldsymbol{x}_i$ 在方向为 $\boldsymbol{w}$ 的直线上的投影. 实际上 $\boldsymbol{w}$ 的绝对值是无关紧要的, 它只是使 y_k 乘上一个常数比例因子, 重要的是选择 $\boldsymbol{w}$ 的方向. $\boldsymbol{w}$ 的方向不同, 将使样本投影后的可分离程度不同, 从而直接影响判别效果. 因此, 寻找最好投影方向的问题, 在数学上就是寻找最佳的变换向量 $\boldsymbol{w}^*$ 的问题.

下面研究如何得到最佳 $\boldsymbol{w}$ 方向的解析表式. 先定义几个必要的基本参量.

(1) 在 n 维 $\boldsymbol{X}$ 特征空间

(a) 各类样本均值向量 $\boldsymbol{m}_k$(n 维向量)

$$\boldsymbol{m}_{\mathrm{k}} = \frac{1}{N_k}\sum_{\boldsymbol{x}_i\in\boldsymbol{X}_k}\boldsymbol{x}_i, \quad k=1,2 \tag{3.2.2}$$

(b) 样本类内离散度矩阵 $\boldsymbol{S}_{\mathrm{k}}$ 和总类内离散度矩阵 $\boldsymbol{S}_{\mathrm{w}}$($n\times n$ 矩阵)

$$\boldsymbol{S}_{\mathrm{k}} = \frac{1}{N_k}\sum_{\boldsymbol{x}_i\in\boldsymbol{X}_k}(\boldsymbol{x}_i-\boldsymbol{m}_k)(\boldsymbol{x}_i-\boldsymbol{m}_k)^{\mathrm{T}}, \qquad k=1,2 \tag{3.2.3}$$

$$\boldsymbol{S}_{\mathrm{w}} = \boldsymbol{S}_1 + \boldsymbol{S}_2. \tag{3.2.4}$$

(c) 样本类间离散度矩阵 $\boldsymbol{S}_\mathrm{b}(n \times n$ 矩阵)

$$\boldsymbol{S}_\mathrm{b} = (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^\mathrm{T}. \tag{3.2.5}$$

(2) 在一维 Y 空间

(a) 各类样本均值 $\tilde{m}_\mathrm{k}$

$$\tilde{m}_\mathrm{k} = \frac{1}{N_k} \sum_{y_i \in Y_k} y_i, \qquad k = 1, 2 \tag{3.2.6}$$

(b) 样本类内离散度 $\tilde{S}_\mathrm{k}^2$ 和总类内离散度 $\tilde{S}_\mathrm{w}^2$

$$\tilde{S}_\mathrm{k}^2 = \sum_{y_i \in Y_\mathrm{k}} (y_i - \tilde{m}_\mathrm{k})^2, \qquad k = 1, 2 \tag{3.2.7}$$

$$\tilde{S}_\mathrm{w}^2 = \tilde{S}_1^2 + \tilde{S}_2^2. \tag{3.2.8}$$

我们希望经过投影后, 在一维 Y 空间内不同类的样本尽可能分离得开些, 即两类均值之差 $(\tilde{m}_1 - \tilde{m}_2)$ 越大越好; 同时希望各类样本内部尽量密集, 即类内离散度 $\tilde{S}_\mathrm{k}^2(k = 1, 2)$ 越小越好. 因此, 我们可以定义 Fisher 准则函数为

$$J_\mathrm{F}(\boldsymbol{w}) = \frac{(\tilde{m}_1 - \tilde{m}_2)^2}{\tilde{S}_1^2 + \tilde{S}_2^2}. \tag{3.2.9}$$

显然, 应寻找使 $J_\mathrm{F}(\boldsymbol{w})$ 尽可能大的 $\boldsymbol{w}$ 作为投影方向. 但上式中的 $J_\mathrm{F}(\boldsymbol{w})$ 并不显含 $\boldsymbol{w}$, 因此必须设法将 $J_\mathrm{F}(\boldsymbol{w})$ 写成 $\boldsymbol{w}$ 的显函数形式. 由式 (3.2.6) 可推出

$$\tilde{m}_\mathrm{k} = \frac{1}{N_k} \sum_{y_i \in Y_k} y_i = \frac{1}{N_k} \sum_{\boldsymbol{x}_i \in \boldsymbol{X}_k} \boldsymbol{w}^\mathrm{T} \boldsymbol{x}_i = \boldsymbol{w}^\mathrm{T} \left(\frac{1}{N_k} \sum_{\boldsymbol{x}_i \in \boldsymbol{X}_k} \boldsymbol{x}_i \right) = \boldsymbol{w}^\mathrm{T} \boldsymbol{m}_\mathrm{k}, \tag{3.2.10}$$

这样式 (3.2.9) 的分子便可写为

$$\begin{aligned}(\tilde{m}_1 - \tilde{m}_2)^2 &= (\boldsymbol{w}^\mathrm{T}\boldsymbol{m}_1 - \boldsymbol{w}^\mathrm{T}\boldsymbol{m}_2)^2 \\ &= \boldsymbol{w}^\mathrm{T}(\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^\mathrm{T}\boldsymbol{w} = \boldsymbol{w}^\mathrm{T}\boldsymbol{S}_\mathrm{b}\boldsymbol{w}.\end{aligned} \tag{3.2.11}$$

再考察 $J_\mathrm{F}(\boldsymbol{w})$ 的分母与 $\boldsymbol{w}$ 的关系. 把式 (3.2.1) 和式 (3.2.10) 代入式 (3.2.7) 可得

$$\begin{aligned}\tilde{S}_\mathrm{k}^2 &= \sum_{y_i \in Y_k} (y_i - \tilde{m}_\mathrm{k})^2 = \sum_{\boldsymbol{x}_i \in \boldsymbol{X}_k} \left(\boldsymbol{w}^\mathrm{T}\boldsymbol{x}_i - \boldsymbol{w}^\mathrm{T}\boldsymbol{m}_\mathrm{k}\right)^2 \\ &= \boldsymbol{w}^\mathrm{T} \left[\sum_{\boldsymbol{x}_i \in \boldsymbol{X}_k} (x_i - \boldsymbol{m}_\mathrm{k})(x_i - \boldsymbol{m}_\mathrm{k})^\mathrm{T} \right] \boldsymbol{w} = \boldsymbol{w}^\mathrm{T}\boldsymbol{S}_\mathrm{k}\boldsymbol{w},\end{aligned}$$

因此

$$\tilde{S}_1^2+\tilde{S}_2^2=\boldsymbol{w}^{\mathrm{T}}\left(\boldsymbol{S}_1+\boldsymbol{S}_2\right)\boldsymbol{w}=\boldsymbol{w}^{\mathrm{T}}S_{\mathrm{w}}\boldsymbol{w}. \tag{3.2.12}$$

将式 (3.2.11) 和 (3.2.12) 代入式 (3.2.9) 可得 $J_{\mathrm{F}}(\boldsymbol{w})$ 的 $\boldsymbol{w}$ 显函数形式:

$$J_{\mathrm{F}}\left(\boldsymbol{w}\right)=\frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{b}}\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{w}}\boldsymbol{w}}. \tag{3.2.13}$$

使 $J_{\mathrm{F}}(\boldsymbol{w})$ 取极大值时的 $\boldsymbol{w}^*$ 是最佳的投影方向, 因此需要求使 $J_{\mathrm{F}}(\boldsymbol{w})$ 取极大值时的 $\boldsymbol{w}^*$. 上式中的 $J_{\mathrm{F}}(\boldsymbol{w})$ 是著名的广义 Rayleigh 商, 可以用 Lagrange 乘子法求解它的极大值点. 令分母等于非零常数, 即令

$$\boldsymbol{w}^{\mathrm{T}}S_{\mathrm{w}}\boldsymbol{w}=c\neq 0.$$

定义 Lagrange 函数为

$$L(\boldsymbol{w},\lambda)=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{w}}\boldsymbol{w}-\lambda(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{w}}\boldsymbol{w}-c), \tag{3.2.14}$$

式中, λ 为 Lagrange 乘子. 将式 (3.2.14) 对 $\boldsymbol{w}$ 求偏导数并令其等于 0, 可得

$$\frac{\partial L(\boldsymbol{w},\lambda)}{\partial \boldsymbol{w}}=S_{\mathrm{b}}\boldsymbol{w}-\lambda S_{\mathrm{w}}\boldsymbol{w}=0,$$

于是有

$$S_{\mathrm{b}}\boldsymbol{w}^*=\lambda\boldsymbol{S}_{\mathrm{w}}\boldsymbol{w}^*, \tag{3.2.15}$$

其中, $\boldsymbol{w}^*$ 就是 $J_{\mathrm{F}}(\boldsymbol{w})$ 的极值解.

由式 (3.2.3)~(3.2.4), $\boldsymbol{S}_{\mathrm{w}}$ 正比于 n 维特征空间内的样本协方差矩阵, 它是对称的和半正定的, 当样本数目 $N>n$ 时通常是非奇异的, 所以可有

$$\boldsymbol{S}_{\mathrm{w}}^{-1}\boldsymbol{S}_{\mathrm{b}}\boldsymbol{w}^*=\lambda\boldsymbol{w}^*. \tag{3.2.16}$$

求解式 (3.2.16) 就是求一般矩阵 $\boldsymbol{S}_{\mathrm{w}}^{-1}S_{\mathrm{b}}$ 的本征值和本征向量问题. 但在我们的问题中, 利用式 (3.2.5)$\boldsymbol{S}_{\mathrm{b}}$ 的定义, 式 (3.2.16) 左边 $\boldsymbol{S}_{\mathrm{b}}\boldsymbol{w}^*$ 可写成

$$\boldsymbol{S}_{\mathrm{b}}\boldsymbol{w}^*=(\boldsymbol{m}_1-\boldsymbol{m}_2)(\boldsymbol{m}_1-\boldsymbol{m}_2)^{\mathrm{T}}\boldsymbol{w}^*=(\boldsymbol{m}_1-\boldsymbol{m}_2)R,$$

式中

$$R=(\boldsymbol{m}_1-\boldsymbol{m}_2)^{\mathrm{T}}\boldsymbol{w}^*$$

为一标量, 所以 $\boldsymbol{S}_{\mathrm{b}}\boldsymbol{w}^*$ 总是与向量 $(\boldsymbol{m}_1-\boldsymbol{m}_2)$ 有相同的方向. 代入式 (3.2.16) 得

$$\lambda\boldsymbol{w}^*=\boldsymbol{S}_{\mathrm{w}}^{-1}\boldsymbol{S}_{\mathrm{b}}\boldsymbol{w}^*=\boldsymbol{S}_{\mathrm{w}}^{-1}(\boldsymbol{m}_1-\boldsymbol{m}_2)R.$$

于是有

$$\boldsymbol{w}^* = \boldsymbol{S}_{\mathrm{w}}^{-1}(\boldsymbol{m}_1 - \boldsymbol{m}_2)R/\lambda. \tag{3.2.17}$$

由于我们的目的是寻找最佳投影方向, $\boldsymbol{w}^*$ 的比例因子并不重要, 因此可以忽略比例因子 R/λ, 得到 $\boldsymbol{w}^*$ 的表式:

$$\boldsymbol{w}^* = \boldsymbol{S}_{\mathrm{w}}^{-1}(\boldsymbol{m}_1 - \boldsymbol{m}_2). \tag{3.2.18}$$

$\boldsymbol{w}^*$ 就是使 Fisher 准则函数 $J_{\mathrm{F}}(\boldsymbol{w})$ 取极大值时的解, 也就是 n 维 $\boldsymbol{X}$ 特征空间到一维 Y 空间的最佳投影方向. 有了 $\boldsymbol{w}^*$, 就可以按照式 (3.2.1) 把 n 维样本 $\boldsymbol{x}_i$, $i = 1, 2, \cdots, N$ 投影到一维 Y 空间, 这实际上是多维空间到一维空间的一种映射.

这样, 就将 n 维样本的分类问题转化为一维样本的分类问题. 根据两类训练样本 $\boldsymbol{X}_1$ 和 $\boldsymbol{X}_2$ 对应的一维 Y 空间中的投影值 y_i, 容易找到区分两类样本的分界点阈值 y_0, 例如可选择一维 Y 空间中的投影值 y_i 的均值 $\tilde{m}$ 作为阈值:

$$y_0 = \frac{N_1\tilde{m}_1 + N_2\tilde{m}_2}{N_1 + N_2} = \tilde{m}, \tag{3.2.19}$$

或两类样本均值的平均 $\tilde{m}$ 作为阈值:

$$y_0 = \frac{\tilde{m}_1 + \tilde{m}_2}{2}. \tag{3.2.20}$$

于是得到决策规则

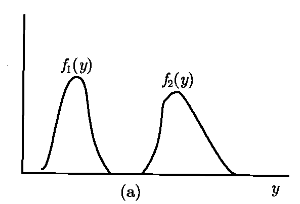
$$\begin{aligned} g(\boldsymbol{x}) &= \boldsymbol{w}^{*\mathrm{T}}\boldsymbol{x} - y_0 \geqslant 0 \qquad \to \boldsymbol{x} \in \omega_1, \\ g(\boldsymbol{x}) &= \boldsymbol{w}^{*\mathrm{T}}\boldsymbol{x} - y_0 < 0 \qquad \to \boldsymbol{x} \in \omega_2. \end{aligned} \tag{3.2.21}$$

对于任意的未知样本 $\boldsymbol{x}$, 只要计算它的投影点 $y = \boldsymbol{w}^{*\mathrm{T}}\boldsymbol{x}$, 就可以按照决策规则式 (3.2.21) 判断它属于什么类别.

由式 (3.2.18) 知, 当两类样本的均值向量相等时 $(\boldsymbol{m}_1 = \boldsymbol{m}_2)$, 找不到最佳投影方向 $\boldsymbol{w}^*$. 即使两类样本的总体分布的形状有很大差异, Fisher 方法仍无法对两类样本作出判别. 在这种情形下, 需要对特征向量作适当的变换 (例如平移, 旋转), 使得 $\boldsymbol{m}_1 \neq \boldsymbol{m}_2$ 成立, 才能利用 Fisher 判别方法.

假定属于模式类 ω_1 和 ω_2 的样本子集 Y_1 和 Y_2 对应的随机变量的概率密度用 $f_1(y)$ 和 $f_2(y)$ 表示. 图 3.5(a) 表示只要选取适当的阈值 y_0, 两类样本可以用 Fisher 线性判别完全正确地区分开来; 而图 3.5(b) 表示无论选取什么阈值 y_0, 两类样本也不可能用 Fisher 线性判别完全正确地区分开来. 这时, 存在误判率的问题.

如第一章所述, 粒子物理实验数据分析的目的是把信号事例从大量本底事例中挑选出来 (称为事例判选), 因此是一个两类模式的判别问题. 一个好的事例判选判



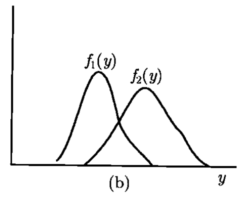


图 3.5 Fisher 线性判别的适用性

(a) 不存在误判; (b) 存在误判

据 (事例分类器) 应当对信号事例有高的选择效率, 有低的误判率 (即对本底事例有低的选择效率或高的排除率). 不失一般性, 假定信号事例样本属于模式类 ω_1, 本底事例样本属于 ω_2. 相应地, 信号和本底的概率密度用 $f_S(y)$ 和 $f_B(y)$ 表示. 对于给定阈值 y_0, 决策规则 $g(\boldsymbol{x}) = \boldsymbol{w}^{*T}\boldsymbol{x} - y_0 > 0$ 正确地选定一个信号事例的效率为 ε_{SS}

$$\varepsilon_{SS} = \int_{y_0}^{\infty} f_S(y)\mathrm{d}y \tag{3.2.22}$$

决策规则 $g(\boldsymbol{x}) = \boldsymbol{w}^{*T}\boldsymbol{x} - y_0 > 0$ 将一个本底事例错误地选择为一个信号事例的误判率为 ε_{SB}

$$\varepsilon_{SB} = \int_{y_0}^{\infty} f_B(y)\mathrm{d}y, \tag{3.2.23}$$

效率与误判率之比也称为信号/本底事例的分辨能力 r

$$r = \varepsilon_{SS}/\varepsilon_{SB}. \tag{3.2.24}$$

尽可能高的信号效率和信号/本底分辨能力, 或者等价地, 尽可能高的信号效率和尽可能低的误判率, 是粒子物理实验中事例分类器的基本要求, 也是一般的模式分类器的基本要求. 当然, 在许多情况下, 这两者是互相矛盾的, 实验者需要根据具体的要求选择适当的阈值 y_0 来达到对于信号效率和误判率的适当折衷.

一般情况下 $f_S(y)$ 和 $f_B(y)$ 是未知或难以求得的. 但如果有了足够数量的信号和本底事例的训练样本, 可以容易地求得 $\varepsilon_{SS}, \varepsilon_{SB}$ 和 r 的估计量及其方差. 假定信号和本底事例的训练样本个数分别为 N_S 和 N_B, 其中用决策规则 $g(\boldsymbol{x}) = \boldsymbol{w}^{*T}\boldsymbol{x} - y_0 > 0$ 选定为信号事例的个数分别为 n_{SS} 和 n_{SB}, 则 $\varepsilon_{SS}, \varepsilon_{SB}$ 和 r 的估计量为

$$\hat{\varepsilon}_{SS} = n_{SS}/N_S$$

$$\hat{\varepsilon}_{SB} = n_{SB}/N_B$$

$$\hat{r} = \frac{\hat{\varepsilon}_{SS}}{\hat{\varepsilon}_{SB}} = \frac{n_{SS}N_B}{n_{SB}N_S}. \tag{3.2.25}$$

这些估计量的方差可以由二项分布求得为

$$
\begin{aligned}
V(\hat{\varepsilon}_{\mathrm{SS}}) &\cong \frac{\hat{\varepsilon}_{\mathrm{SS}}(1-\hat{\varepsilon}_{\mathrm{SS}})}{N_{\mathrm{S}}},\\
V(\hat{\varepsilon}_{\mathrm{SB}}) &\cong \frac{\hat{\varepsilon}_{\mathrm{SB}}(1-\hat{\varepsilon}_{\mathrm{SB}})}{N_{\mathrm{B}}},\\
\frac{V(\hat{r})}{\hat{r}^2} &\cong \frac{V(\hat{\varepsilon}_{\mathrm{SS}})}{\hat{\varepsilon}_{\mathrm{SS}}^2} + \frac{V(\hat{\varepsilon}_{\mathrm{SB}})}{\hat{\varepsilon}_{\mathrm{SB}}^2}
\end{aligned}
\tag{3.2.26}
$$

3.3 感知准则函数

3.3.1 几个基本概念

为了便于后面的叙述, 先介绍几个基本概念.

(1) 线性可分性的概率

在 3.1 节已经阐明, 假设已知一组容量 N 的样本集 $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N$, 其中 $\boldsymbol{y}_i$ 是 $\hat{n} = n+1$ 维增广样本向量, 分别来自模式类 ω_1 和 ω_2. 如果存在一个线性分类器能把每个样本正确分类, 即如果存在一个权向量 $\boldsymbol{v}$, 使得对于任意的 $\boldsymbol{y} \in \omega_1$, 都有 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y} > 0$; 而对于任意的 $\boldsymbol{y} \in \omega_2$, 都有 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y} < 0$, 则称该样本集是线性可分的; 否则称为线性不可分的. 反过来, 如果样本集是线性可分的, 则必定存在一个权向量 $\boldsymbol{v}$, 能把该样本集的每个样本正确地分类.

那么对于容量 N 的样本集, 线性可分的概率有多大.

一般来说, 假设有 N 个 n 维样本, 每个样本点被标明属于模式类 ω_1 或 ω_2. 这 N 个 n 维样本共有 2^N 种可能的二分法, 但其中只有一部分是线性二分法, 即对于它们存在某一个超平面能把属于 ω_1 的样本与属于 ω_2 的样本分割开来. 如果 $N > n$ 时没有 $n+1$ 个样本落入 $n-1$ 维子空间内 (如 $n=3$, 这 N 个 3 维样本没有 4 个样本落在同一个 2 维平面内); 而当 $N < n$ 时没有 2 个或以上的样本落入 $n-2$ 维子空间内, 那么, 这 2^N 种可能的二分法中线性二分法所占的比例, 或者说概率, 可用下式给出:

$$
P(N,n) = \begin{cases} 1, & N \leqslant n, \\ 2^{1-N} \sum\limits_{i=1}^{n} C_{N-1}^{i}, & N > n. \end{cases} \tag{3.3.1}
$$

这一函数表示在图 3.6 中.

(2) 样本的规范化

由前面的讨论可知, 如果样本集 $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N$, 其中 $\boldsymbol{y}_i$ 是 $\hat{n} = n+1$ 维增广样本向量, 分别来自模式类 ω_1 和 ω_2. 如果存在一个线性分类器能把每个样本正确分

类, 即如果存在一个权向量 $\boldsymbol{v}$, 使得对于任意的 $\boldsymbol{y} \in \omega_1$, 都有 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y} > 0$; 而对于任意的 $\boldsymbol{y} \in \omega_2$, 都有 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y} < 0$, 则称该样本集是线性可分的; 否则称为线性不可分的.

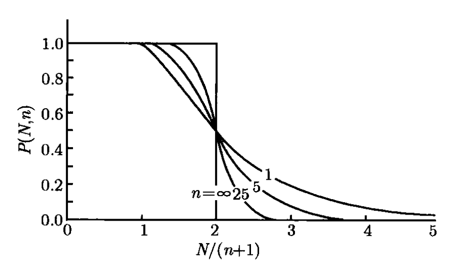


图 3.6 n 维空间 N 个样本点的二分法可以线性分割的比例

由前面的讨论可知, 如果样本集 $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N$ 是线性可分的, 则必定存在一个或一个以上的权向量 $\boldsymbol{v}$, 使得

$$\begin{cases} \boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}_i > 0, & \text{对一切 } \boldsymbol{y}_i \in \omega_1 \\ \boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}_j < 0, & \text{对一切 } \boldsymbol{y}_j \in \omega_2 \end{cases} \tag{3.3.2}$$

或者说, 满足上式的一切权向量 $\boldsymbol{v}$ 都能将全部 N 个样本正确地分类. 上式中如果在属于 ω_2 类的样本 $\boldsymbol{y}_j$ 前面加一个负号, 即令 $\boldsymbol{y}'_j = -\boldsymbol{y}_j$, 则有 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}'_j > 0$. 因此, 若令

$$\boldsymbol{y}'_m = \begin{cases} \boldsymbol{y}_i, & \text{对一切 } \boldsymbol{y}_i \in \omega_1 \\ -\boldsymbol{y}_j, & \text{对一切 } \boldsymbol{y}_j \in \omega_2 \end{cases} \tag{3.3.3}$$

那么, 我们可以不管样本原来的类别标志, 只要寻找一个对全部 N 个样本的 $\boldsymbol{y}'_m$ 都满足 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}'_m > 0$, $m = 1, 2, \cdots, N$ 的权向量 $\boldsymbol{v}$ 就可以了. 上述过程称为样本的规范化, $\boldsymbol{y}'_m$ 叫做规范化增广样本向量. 在后面我们仍然用 $\boldsymbol{y}_m$ 来表示它.

(3) 解向量和解区

在线性可分的情形下, 满足 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}_m > 0$, $m = 1, 2, \cdots, N$ 的权向量 $\boldsymbol{v}$ 称为解向量, 记为 $\boldsymbol{v}^*$. 权向量 $\boldsymbol{v}$ 可以理解为权空间中的一个点, 每个样本 $\boldsymbol{y}_m$ 对 $\boldsymbol{v}$ 的可能位置都起到限制作用, 即要求 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}_m > 0$. 方程 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}_m > 0$ 确定了一个通过权空间原点的超平面 $\hat{H}_m$, 其法向量为 $\boldsymbol{y}_m$. 解向量如果存在, 则必定在超平面 $\hat{H}_m$ 的正侧, 因为只有在正侧才能满足 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}_m > 0$. N 个样本将产生 N 个超平面, 每个超平面把权空间分为两个半空间. 所以, 解向量如果存在, 必定落在 N 个正半空间的交叠区, 而且该区中的任意向量都是解向量 $\boldsymbol{v}^*$. 该区域称为权向量 $\boldsymbol{v}$ 的解区. 图 3.7 是二维情况下解区的图示.

(4) 对解区的限制

对解区加以限制的目的在于使解向量 $\boldsymbol{v}^*$ 更可靠. 通常认为, 越靠近解区中间的解向量越能对待定的新样本正确分类. 因此, 可引入余量 $b>0$, 并寻找满足 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}_m \geqslant b$ 的解向量 $\boldsymbol{v}^*$. 显然, 由 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}_m \geqslant \boldsymbol{b} > 0$ 得到的正半空间的交叠区 (即新解区) 位于原解区之中, 而且它的边界离开原解区边界的距离为 $b/\|\boldsymbol{y}_m\|$, 如图 3.8 所示. 实际上, 只要解向量 $\boldsymbol{v}^*$ 的算法不至于收敛到解区的边界, 这样的解向量 $\boldsymbol{v}^*$ 都能满足要求. 显然, 通过引入余量 $b>0$ 可以很好地解决这一问题.

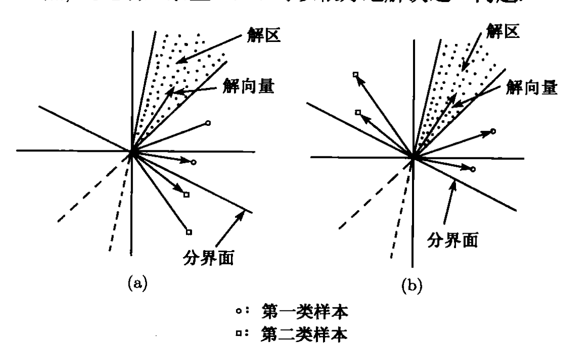


图 3.7　权向量的解区和解向量的示意图

(a) 未规范化样本; (b) 规范化样本

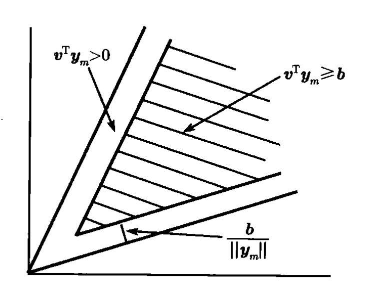


图 3.8　引入余量的权向量的解区

3.3.2　感知准则函数

设有一组样本 $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N$, 其中 $\boldsymbol{y}_m$, $m=1,2,\cdots,N$ 是规范化增广样本向量. 我们的目的是寻找一个解向量 $\boldsymbol{v}^*$, 使得

$$\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}_m > 0, \qquad m=1,2,\cdots,N.$$

显然, 仅当样本是线性可分的情况下, 问题才有解.

因此, 这里考虑的是处理线性可分问题的算法. 先构造一个如下的准则函数

$$J_{\mathrm{P}}(\boldsymbol{v}) = \sum_{y\in Y_{\mathrm{k}}} (-\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}), \tag{3.3.4}$$

式中, Y_{k} 是被权向量 $\boldsymbol{v}$ 错误分类的样本集合. 当样本 $\boldsymbol{y}$ 被错误分类时, 就有 $-\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}_m \geqslant 0$, 因此, $J_{\mathrm{P}}(\boldsymbol{v})$ 总是大于等于 0, 而且仅当 $\boldsymbol{v}$ 为解向量或 $\boldsymbol{v}$ 在解区边界上时 $J_{\mathrm{P}}(\boldsymbol{v})$ 才等于 0. 也就是说, 当且仅当 Y_{k} 为空集时,

$$J_{\mathrm{P}}^{*}(\boldsymbol{v}) = \min J_{\mathrm{P}}(\boldsymbol{v}) = 0. \tag{3.3.5}$$

这时将不存在错分样本, 这里的权向量 $\boldsymbol{v}$ 就是我们要寻找的解向量 $\boldsymbol{v}^*$. 这一准则函数是 20 世纪 50 年代 Rosenblatt 提出, 试图用于人工神经网络的脑模型感知器 (参见第五章) 上的, 故一般称为感知准则函数.

由于准则函数 $J_{\mathrm{P}}(\boldsymbol{v})$ 极小时对应的 $\boldsymbol{v}$ 为解向量 $\boldsymbol{v}^*$, 问题就转化为求准则函数 $J_{\mathrm{P}}(\boldsymbol{v})$ 的极小值时的 $\boldsymbol{v}$. 这可以由一般的最优化计算方法来达到. 例如可以采用梯度下降法, 将式 (3.3.5) 对 $\boldsymbol{v}$ 求梯度, 有

$$\nabla J_{\mathrm{P}}(\boldsymbol{v}) = \frac{\partial J_{\mathrm{P}}(\boldsymbol{v})}{\partial \boldsymbol{v}} = \sum_{y\in Y_{\mathrm{k}}} (-\boldsymbol{y}), \tag{3.3.6}$$

梯度下降法的迭代公式为

$$\boldsymbol{v}(k+1) = \boldsymbol{v}(k) - \rho_k \nabla J.$$

将式 (3.3.6) 代入上式得到可以用来作实际运算的迭代公式

$$\boldsymbol{v}(k+1) = \boldsymbol{v}(k) + \rho_k \sum_{y\in Y_{\mathrm{k}}} \boldsymbol{y}, \tag{3.3.7}$$

式中, Y_{k} 是被权向量 $\boldsymbol{v}(k)$ 错分的样本集合.

梯度下降法可以简单地表述为, 任意给定初始权向量 $\boldsymbol{v}(1)$, 第 $k+1$ 次迭代时的权向量 $\boldsymbol{v}(k+1)$ 等于第 k 次迭代时的权向量 $\boldsymbol{v}(k)$ 加上被 $\boldsymbol{v}(k)$ 错分的样本值之和乘以某个系数 ρ_k. 可以证明, 对于线性可分的样本集, 经过对初始权向量 $\boldsymbol{v}(1)$ 的有限次迭代修正, 一定可以找到一个解向量 $\boldsymbol{v}^*$, 即迭代算法在有限次迭代后收敛, 其收敛速度的快慢取决于初始权向量 $\boldsymbol{v}(1)$ 和系数 ρ_k.

上述梯度下降法可以加以简化. 从式 (3.3.7) 可以看出, 在每次迭代中, 只有那些被错分的样本才对权向量 $\boldsymbol{v}$ 的修正起作用. 因此可以将样本集看作一个不断出现的样本序列, 逐个样本考虑对权向量 $\boldsymbol{v}$ 的修正. 对于任意权向量 $\boldsymbol{v}(k)$, 如果它把某个样本错分了, 则对 $\boldsymbol{v}(k)$ 作一次修正, 这种方法称为单样本修正法. 例如样本集

容量 N=3, 考虑由 3 个样本组成的序列 $\tilde{\boldsymbol{y}}_1, \boldsymbol{y}_2, \tilde{\boldsymbol{y}}_3, \boldsymbol{y}_1, \tilde{\boldsymbol{y}}_2, \boldsymbol{y}_3, \tilde{\boldsymbol{y}}_1, \boldsymbol{y}_2, \boldsymbol{y}_3 \cdots$, 其中有 "~" 记号的样本表示被错分的样本, 首先, 把错分样本的序列 $\tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_3, \tilde{\boldsymbol{y}}_2, \tilde{\boldsymbol{y}}_1, \cdots$ 重新记为 $\tilde{\boldsymbol{y}}^1, \tilde{\boldsymbol{y}}^2, \tilde{\boldsymbol{y}}^3, \tilde{\boldsymbol{y}}^4, \cdots$; 其次, 把系数 ρ_k 看作不随 k 而变化的常数, 不失一般性, 可令 $\rho_k = 1$. 这样简化后的梯度下降法可以表示为

$$\begin{cases} \boldsymbol{v}(1), & \text{任意} \\ \boldsymbol{v}(k+1) = \boldsymbol{v}(k) + \tilde{\boldsymbol{y}}^k \end{cases} \tag{3.3.8}$$

其中, $\boldsymbol{y}^k$ 是被 $\boldsymbol{v}(k)$ 错分的样本. 这样的迭代一直进行到对于原样本集 $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N$ 的一次循环中不再出现被错分的样本为止, 就得到解向量 $\boldsymbol{v}^*$.

算法式 (3.3.8) 称为固定增量法, 它首先由 Rosenblatt 提出, 并证明了其收敛性 (这里从略), 称为感知收敛定理.

3.4 最小错分样本数准则函数

3.3 节已经指出, 感知准则函数及其梯度下降算法只适用于样本集线性可分的情形, 对于样本集线性不可分的情形, 迭代过程永远不会终结, 即算法不收敛. 在实际问题中, 往往事先无法知道样本集是否线性可分. 因此我们希望能找到一种既适用于样本集线性可分、也适用于样本集线性不可分情况的算法. 这种算法对于线性可分问题应当可以得到一个如感知准则函数那样的解向量 $\boldsymbol{v}^*$, 使得对两类样本集的所有样本能正确地分类; 而对于线性不可分问题, 则能得到一个使两类样本集被错分的样本数达到极小的解向量 $\boldsymbol{v}^*$. 上述准则称为最小错分样本数准则.

设有一组样本 $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N$, 其中 $\boldsymbol{y}_m$, $\boldsymbol{m} = 1, 2, \cdots, N$ 是规范化增广样本向量. 如果存在权向量 $\boldsymbol{v}^*$, 使得

$$\boldsymbol{v}^{\mathrm{T}} \boldsymbol{y}_m > 0, \quad \boldsymbol{m} = 1, 2, \cdots, N. \tag{3.4.1}$$

即式 (3.4.1) 所示的 N 个线性不等式有解, 即不等式组相一致, 则样本集 $\boldsymbol{y}_m$ 是线性可分的, 并被其解向量 $\boldsymbol{v}^*$ 正确分类. 若不等式组无解, 即不等式组不一致, 样本集 $\boldsymbol{y}_m$ 线性不可分, 则对于任何权向量 $\boldsymbol{v}$, 必定有某些样本被错误地分类. 这时我们只能寻找使得不等式得到满足的数目最大的权向量 $\boldsymbol{v}$, 把它作为问题的解 $\boldsymbol{v}^*$. 先用矩阵形式重写式 (3.4.1) 所示的不等式组:

$$\boldsymbol{Y}\boldsymbol{v} > 0 \tag{3.4.2}$$

其中

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_1^{\mathrm{T}} \\ \boldsymbol{y}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{y}_N^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1\hat{n}} \\ y_{21} & y_{22} & \cdots & y_{2\hat{n}} \\ \vdots & \vdots & & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{N\hat{n}} \end{bmatrix}, \tag{3.4.3}$$

$\boldsymbol{Y}$ 是 $N \times \hat{n}$ 规范化增广样本矩阵, $\hat{n}$ 是样本 $\boldsymbol{y}_m$ 的维数. 为了使解更可靠, 引入余量 $\boldsymbol{b} > 0$, 上式改写为

$$\boldsymbol{Y}\boldsymbol{v} \geqslant \boldsymbol{b} > 0. \tag{3.4.4}$$

不失一般性, 可以取

$$\boldsymbol{b} = \left.\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}\right\} N \text{ 个 } 1.$$

对于式 (3.4.4), 可以定义准则函数

$$J_{q1}(\boldsymbol{v}) = \|(\boldsymbol{Y}\boldsymbol{v} - \boldsymbol{b}) - |\boldsymbol{Y}\boldsymbol{v} - \boldsymbol{b}|\|^2. \tag{3.4.5}$$

准则函数 $J_{q1}(\boldsymbol{v})$ 中如果 $\boldsymbol{Y}\boldsymbol{v} > \boldsymbol{b}$, 则 $\boldsymbol{Y}\boldsymbol{v} - b$ 与 $|\boldsymbol{Y}\boldsymbol{v} - b|$ 同号, 故 $J_{q1}(\boldsymbol{v}) = 0$; 反之, 如果有某些 $\boldsymbol{y}_m$ 不满足 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}_m > \boldsymbol{b}_m$, 则 $\boldsymbol{Y}\boldsymbol{v}_m - \boldsymbol{b}_m$ 与 $|\boldsymbol{Y}\boldsymbol{v}_m - \boldsymbol{b}_m|$ 异号, 因此 $J_{q1}(\boldsymbol{v}) > 0$. 不满足 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}_m > \boldsymbol{b}_m$ 的样本 $\boldsymbol{y}_m$ 数量越多, $J_{q1}(\boldsymbol{v})$ 越大. 显然 $J_{q1}(\boldsymbol{v})$ 取极小值时的 $\boldsymbol{v}$ 为问题的最优解 $\boldsymbol{v}^*$; 并且当样本集 $\boldsymbol{y}_m$ 是线性可分时 $J_{q1}(\boldsymbol{v}) = 0$, 当样本集 $\boldsymbol{y}_m$ 是线性不可分时 $J_{q1}(\boldsymbol{v}) > 0$. 准则函数 $J_{q1}(\boldsymbol{v})$ 求极小的问题可由最优化方法求解, 这里不再讨论.

式 (3.4.5) 表示的准则函数 $J_{q1}(\boldsymbol{v})$, 在不等式组不一致的情况下, 对某些样本可能存在 $0 < \boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}_m < \boldsymbol{b}_m$. 这时因为 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}_m > 0$, $\boldsymbol{y}_m$ 应该能被正确分类; 但又由于 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}_m < \boldsymbol{b}_m$, 所以用式 (3.4.5) 准则函数 $J_{q1}(\boldsymbol{v})$ 得到的解 $\boldsymbol{v}^*$ 来分类时该 $\boldsymbol{y}_m$ 会被错分. 因此需要对式 (3.4.5) 表示的准则函数 $J_{q1}(\boldsymbol{v})$ 作适当的修正.

如果式 (3.4.5) 中取 $\boldsymbol{b} = 0$, 则准则函数变成

$$J_{q1}(\boldsymbol{v}) = \|\, \boldsymbol{Y}\boldsymbol{v} - |\boldsymbol{Y}\boldsymbol{v}|\, \|^2. \tag{3.4.6}$$

在一致的情况下, 利用最优化方法求上述准则函数的极小可收敛于 $\boldsymbol{Y}\boldsymbol{v} > 0$ 的解向量 $\boldsymbol{v}^*$. 在不一致的情形下, 由于 $J_{q1}(\boldsymbol{v})$ 是严格的凸函数, 其唯一的极小点是 $\boldsymbol{v} = 0$, 而且有 $J_{q1}(\boldsymbol{v}) = 0$. 因此得不到解向量 $\boldsymbol{v}^*$. 在这种情况下, 我们可以用

$$F(\boldsymbol{v}) = \frac{\|\, \boldsymbol{Y}\boldsymbol{v} - |\boldsymbol{Y}\boldsymbol{v}|\, \|^2}{\|\boldsymbol{v}\|^2} \tag{3.4.7}$$

作为准则函数来克服上述困难. 当利用 $\nabla_{\boldsymbol{v}}F(\boldsymbol{v}) = 0$ 求 $F(\boldsymbol{v})$ 的极小时, 容易得到下述关系:

$$\nabla_v F(\boldsymbol{v}) \propto \nabla_v J_{q1}(\boldsymbol{v}) - 2F(\boldsymbol{v}) \cdot \boldsymbol{v} = 0. \tag{3.4.8}$$

这说明, 使 $F(\boldsymbol{v})$ 达到极小与 $\boldsymbol{v} \neq 0$ 并满足

$$\nabla_{\boldsymbol{v}} J_{q1}(\boldsymbol{v}) = 2F(\boldsymbol{v}) \cdot \boldsymbol{v} \tag{3.4.9}$$

条件下使 $J_{q1}(\boldsymbol{v})$ 达到极小是等价的. 这样得到的权向量 $\boldsymbol{v}$ 就是问题的解向量 $\boldsymbol{v}^*$.

3.5 最小平方误差准则函数

3.5.1 平方误差准则函数及其 MSE 解

3.4 节已经指出, 最小错分样本数准则是寻找一个权向量 $\boldsymbol{v}$, 使得不等式 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}_m > 0$ 得以满足的样本 $\boldsymbol{y}_m$ 的数目最大, 从而使错分样本数最少. 在不等式组一致的情形下, 则得到解区中的一个解向量 $\boldsymbol{v}^*$.

现在我们把不等式组变为如下形式:

$$\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}_m = b_m > 0$$

其中, b_m 为任意给定的正常数. 将上式写成联立方程组的形式即为

$$\boldsymbol{Y}\boldsymbol{v} = \boldsymbol{b} \tag{3.5.1}$$

其中, $\boldsymbol{Y}$ 是 $N \times \hat{n}$ 规范化增广样本矩阵, 由式 (3.4.3) 给定, $\boldsymbol{b}$ 是 N 维向量:

$$\boldsymbol{b} = (b_1, \quad b_2, \quad \cdots \quad b_N)^{\mathrm{T}}$$

$b_m > 0, \ m = 1, 2, \cdots, N.$

通常样本数 N 总是大于维数 $\hat{n}$, 因此 $\boldsymbol{Y}$ 是长方阵, 且一般为列满秩阵. 这对应于方程个数多于未知数的情况, 因此一般为矛盾方程组, 通常不存在精确解. 我们可以定义一个误差向量

$$\boldsymbol{e} = \boldsymbol{Y}\boldsymbol{v} - \boldsymbol{b}$$

并定义平方误差准则函数

$$J_{\mathrm{s}}(\boldsymbol{v}) = \|\boldsymbol{e}\|^2 = \|\boldsymbol{Y}\boldsymbol{v} - \boldsymbol{b}\|^2 = \sum_{m=1}^{N} (\boldsymbol{v}_m^{\mathrm{T}}\boldsymbol{y} - b_m)^2. \tag{3.5.2}$$

寻找一个使 $J_{\mathrm{s}}(\boldsymbol{v})$ 达到极小的权向量 $\boldsymbol{v}$ 作为问题的解, 这就是矛盾方程组的最小二乘近似解, 也称为伪逆解或 MSE 解, 我们仍用 $\boldsymbol{v}^*$ 表示. 式 (3.5.2) 定义的准则函数也称 MSE 准则函数.

现在来求 MSE 解的显著表式. 对 $J_s(\boldsymbol{v})$ 求梯度, 得

$$\nabla J_s(\boldsymbol{v}) = \sum_{m=1}^{N} 2(\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}_m - b_m)\boldsymbol{y}_m = 2\boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{Y}\boldsymbol{v} - \boldsymbol{b}). \tag{3.5.3}$$

令 $\nabla J_s(\boldsymbol{v}) = 0$ 求极小, 得

$$\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{v}^* = \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{b}. \tag{3.5.4}$$

这样, 求解 $\boldsymbol{Y}\boldsymbol{v} = \boldsymbol{b}$ 的问题转化为求解 $\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{v}^* = \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{b}$ 的问题了. 式 (3.5.4) 的好处是 $\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}$ 是 $\hat{n} \times \hat{n}$ 方阵, 而且一般是非奇异的, 因此可唯一地求得解向量 $\boldsymbol{v}^*$:

$$\boldsymbol{v}^* = (\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y})^{-1}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{b} = \boldsymbol{Y}^{+}\boldsymbol{b}, \tag{3.5.5}$$

式中, $\hat{n} \times N$ 矩阵 $\boldsymbol{Y}^{+}$ 是 $\boldsymbol{Y}$ 的左逆矩阵

$$\boldsymbol{Y}^{+} = (\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y})^{-1}\boldsymbol{Y}^{\mathrm{T}}. \tag{3.5.6}$$

由式 (3.5.4) 知问题的解 $\boldsymbol{v}^*$ 取决于给定的 N 维向量 b, 因此就有一个 $\boldsymbol{b}$ 如何选取的问题. 可以证明, 对于二类问题, N_1 个样本属于 ω_1 类, N_2 个样本属于 ω_2 类, 总样本数 $N = N_1 + N_2$, 当取

$$\boldsymbol{b} = \begin{bmatrix} N/N_1 \\ \vdots \\ N/N_1 \\ N/N_2 \\ \vdots \\ N/N_2 \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} N_1 \text{ 个} \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} N_2 \text{ 个} \end{matrix} \tag{3.5.7}$$

则 MSE 解 $\boldsymbol{v}^*$ 等价于 Fisher 解. 并得到

$$w_0^* = -\boldsymbol{m}^{\mathrm{T}}\boldsymbol{w}^* \tag{3.5.8}$$

和如下决策规则:

$$\begin{aligned} &\boldsymbol{w}^{*\mathrm{T}}(\boldsymbol{x} - \boldsymbol{m}) > 0, \qquad \text{则} \quad \boldsymbol{x} \in \omega_1, \\ &\boldsymbol{w}^{*\mathrm{T}}(\boldsymbol{x} - \boldsymbol{m}) < 0, \qquad \text{则} \quad \boldsymbol{x} \in \omega_2, \end{aligned} \tag{3.5.9}$$

其中, $\boldsymbol{m}$ 是总的样本均值

$$\boldsymbol{m} = \frac{N_1\boldsymbol{m}_1 + N_2\boldsymbol{m}_2}{N}.$$

这与 Fisher 线性判别方法中取 $y_0 = \tilde{m}$ 的情况是相同的.

这里, 我们不加证明地给出 MSE 解的另一个有用性质: 当样本数 N 趋于无穷时, 如果令 $\boldsymbol{b}=\boldsymbol{u}_N$

$$\boldsymbol{u}_N=\left.\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}\right\}N\text{ 个}, \tag{3.5.10}$$

则 MSE 解以最小均方误差逼近贝叶斯判别函数 (参见式 (2.1.22))

$$g_{\mathrm{B}}(\boldsymbol{x})=q(\omega_1|\boldsymbol{x})-q(\omega_2|\boldsymbol{x}) \tag{3.5.11}$$

对问题的解.

3.5.2 MSE 准则函数的梯度下降算法

在计算权向量的 MSE 解

$$\boldsymbol{v}^*=\boldsymbol{Y}^+\boldsymbol{b}$$

时需要计算 $\boldsymbol{Y}$ 的左逆矩阵 $\boldsymbol{Y}^+=(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y})^{-1}\boldsymbol{Y}^{\mathrm{T}}$. 这会带来两个问题: 第一是要求 $(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y})$ 为非奇异矩阵; 第二是求 $\boldsymbol{Y}^+$ 的计算量比较大, 因为 $\boldsymbol{Y}$ 是一个 $N\times\hat{n}$ 矩阵, 而总样本数 N 往往很大, 同时在大量的计算中还可能引入较大的计算误差. 因此在实际工作中往往不用这种解析方法, 而是采用最优化技术如梯度下降法来求解.

由式 (3.5.3) 知 $J_{\mathrm{s}}(\boldsymbol{v})$ 的梯度为

$$\nabla J_{\mathrm{s}}(\boldsymbol{v})=2\boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{Y}\boldsymbol{v}-\boldsymbol{b})$$

则梯度下降算法可表示为

$$\begin{cases}\boldsymbol{v}(1), & \text{任意}\\ \boldsymbol{v}(k+1)=\boldsymbol{v}(k)-\rho_k\boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{Y}\boldsymbol{v}-\boldsymbol{b})\end{cases} \tag{3.5.12}$$

可以证明, 如果选择

$$\rho_k=\rho_1/k, \qquad \rho_1\text{ 任意正常数} \tag{3.5.13}$$

则用该算法得到的权向量序列收敛于使

$$\nabla J_{\mathrm{s}}(\boldsymbol{v})=2\boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{Y}\boldsymbol{v}-\boldsymbol{b})=0$$

的权向量 $\boldsymbol{v}^*$, 即 MSE 解. 无论矩阵 $(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y})$ 是否奇异, 该算法总能产生一个有用的权向量, 而且该算法只计算矩阵与向量的乘积, 避免了 $\boldsymbol{Y}^+=(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y})^{-1}\boldsymbol{Y}^{\mathrm{T}}$ 中的矩阵与矩阵间的乘积运算和矩阵的求逆运算, 大大减小了计算量.

为了进一步减小计算量和存储量, 类似于 3.3.2 小节感知准则函数中介绍的单样本修正法那样, 可以把样本看成一个无限重复出现的序列而逐个样本加以考虑. 这样, 式 (3.5.12) 的算法可修改为

$$\begin{cases} \boldsymbol{v}(1), & \text{任意} \\ \boldsymbol{v}(k+1) = \boldsymbol{v}(k) + \rho_k (b_k - \boldsymbol{v}(k)^{\mathrm{T}} \boldsymbol{y}^k) \boldsymbol{y}^k \end{cases} \tag{3.5.14}$$

其中, $\boldsymbol{y}^k$ 是使 $\boldsymbol{v}(k)^{\mathrm{T}}\boldsymbol{y}^k \neq b_k$ 的样本. 这样的迭代一直进行到对于原样本集 $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N$ 的一次循环中不再出现被错分的样本即 $\boldsymbol{v}(k)^{\mathrm{T}}\boldsymbol{y}^k = b_k$ 为止, 这时, 迭代停止, 得到解向量 $\boldsymbol{v}^*$.

由于 b_k 是任意给定的正常数, 一般说来, 要使 $\boldsymbol{v}(k)^{\mathrm{T}}\boldsymbol{y}^k = b_k$ 成立几乎是不可能的, 因而上述迭代修正过程永远不会终止, 所以必须让 ρ_k 随着 k 的增大而减小, 以保证收敛. 一般选择 $\rho_k = \rho_1/k$, 此时式 (3.5.14) 的算法收敛于满意的解向量 $\boldsymbol{v}^*$. 该算法是对 MSE 准则采用梯度下降法的一个修正, 通常称为 Widrow-Hoff 算法.

3.5.3 随机 MSE 准则函数及其随机逼近算法

前面讲到的算法都是针对确定性样本集的, 但实际上样本总是随机抽取的, 因此应把每个样本都看作抽自某个总体分布的随机变量, 即样本集是随机样本集. 为此, 我们需要定义一个随机的准则函数, 并用处理随机最优化问题的随机逼近算法来求解.

假设样本是按下述方式独立抽取的, 即先按概率 $\pi(\omega_i)$, 选择一个类别状态, 再按 $p(\boldsymbol{x}|\omega_i)$ 选择一个样本 $\boldsymbol{x}$, 每个样本都有一个类别标志, 用 z 来表示. 对于二类问题有

$$z = \begin{cases} +1, & \text{对于 } \boldsymbol{x} \in \omega_1 \\ -1, & \text{对于 } \boldsymbol{x} \in \omega_2 \end{cases} \tag{3.5.15}$$

这样就得到一个无穷的数据序列 $(\boldsymbol{x}_1, z_1), (\boldsymbol{x}_2, z_2), \cdots, (\boldsymbol{x}_k, z_k), \cdots$ 在 $\boldsymbol{x}$ 已知的情况下, 随机变量 $z(\boldsymbol{x})$ 的条件概率为

$$\begin{cases} q(z=1|\boldsymbol{x}) = q(\omega_1|\boldsymbol{x}) \\ q(z=-1|\boldsymbol{x}) = q(\omega_2|\boldsymbol{x}) \end{cases} \tag{3.5.16}$$

由于 $z(\boldsymbol{x})$ 仅取二值, 所以 $z(\boldsymbol{x})$ 的条件期望为

$$E\left[z(\boldsymbol{x})\right] = \sum_z z q(z|\boldsymbol{x}) = q(\omega_1|\boldsymbol{x}) - q(\omega_2|\boldsymbol{x}) = g_{\mathrm{B}}(\boldsymbol{x}) \tag{3.5.17}$$

上式说明 $z(\boldsymbol{x})$ 的条件期望是贝叶斯判别函数 $g_{\mathrm{B}}(\boldsymbol{x})$ (参见式 (2.1.22)).

我们先回忆一下确定性样本情况下的 MSE 准则函数, 即式 (3.5.2)

$$J_s(\boldsymbol{v}) = \|\boldsymbol{Y}\boldsymbol{v} - \boldsymbol{b}\|^2 = \sum_{m=1}^{N} (\boldsymbol{v}_m^{\mathrm{T}}\boldsymbol{y} - b_m)^2.$$

当取 $\boldsymbol{b} = \boldsymbol{u}_N$ 时, 所得线性判别函数 $\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}$ 以最小均方误差逼近贝叶斯判别函数 $g_{\mathrm{B}}(\boldsymbol{x})$(见 3.5.1 小节).

对于随机样本, 我们可以类似地定义 MSE 准则函数如下：

$$J_{\mathrm{SR}}(\boldsymbol{v}) = E\left[(\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{b})^2\right]$$

当令 $b = z$ 时, 有

$$J_{\mathrm{SR}}(\boldsymbol{v}) = E\left[(\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y} - z)^2\right] \tag{3.5.18}$$

可以证明, 对应于使 $J_{\mathrm{SR}}(\boldsymbol{v})$ 极小化的 $\boldsymbol{v}^*$ 的随机线性判别函数 $g_{\mathrm{SR}}(\boldsymbol{x}) = \boldsymbol{v}^{*\mathrm{T}}\boldsymbol{y}$, 仍然以最小均方误差逼近贝叶斯判别函数 $g_{\mathrm{B}}(\boldsymbol{x})$.

有了随机准则函数 $J_{\mathrm{SR}}(\boldsymbol{v})$, 问题就变成如何求出它的极值解 $\boldsymbol{v}^*$. 求 $J_{\mathrm{SR}}(\boldsymbol{v})$ 对于 $\boldsymbol{v}$ 的梯度并令其等于零, 得

$$\nabla J_{\mathrm{SR}}(\boldsymbol{v}) = 2E\left[\left(\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y} - z(\boldsymbol{x})\right)\boldsymbol{y}\right] = 0$$

从而得到问题的解向量 $\boldsymbol{v}^*$:

$$\boldsymbol{v}^* = E\left[(\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}})\right]^{-1} E\left[z(\boldsymbol{x})\boldsymbol{y}\right] \tag{3.5.19}$$

由上式计算解向量 $\boldsymbol{v}^*$ 是不容易的, 可利用最优化算法中的牛顿法, 其迭代公式为

$$\boldsymbol{v}(k+1) = \boldsymbol{v}(k) - D^{-1}\nabla J$$

式中, D 为准则函数 J 的二阶偏导数矩阵, 这里

$$D = 2E\left[\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}\right]$$

因此, 使 $J_{\mathrm{SR}}(\boldsymbol{v})$ 极小化的牛顿迭代公式为

$$\boldsymbol{v}(k+1) = \boldsymbol{v}(k) + \left[E(\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}})\right]^{-1}\left[E(z - \boldsymbol{v}^{\mathrm{T}}\boldsymbol{y})\boldsymbol{y}\right]. \tag{3.5.20}$$

若用样本估计代替期望值计算, 并利用类似于求样本均值时的迭代算法, 可令

$$R(k+1)^{-1} = R(k)^{-1} + \boldsymbol{y}_k\boldsymbol{y}_k^{\mathrm{T}} \tag{3.5.21}$$

式 (3.5.21) 可看作计算 D 的迭代公式. 可以证明

$$R(k+1) = R(k) - \frac{R(k)\boldsymbol{y}_k\left[R(k)\boldsymbol{y}_k\right]^{\mathrm{T}}}{1 + \boldsymbol{y}_k^{\mathrm{T}}R(k)\boldsymbol{y}_k}. \tag{3.5.22}$$

把上式中的 $R(k+1)^{-1}$ 视为 D^{-1}, 则可得到一种改进的随机逼近迭代算法公式:

$$\boldsymbol{v}(k+1)=\boldsymbol{v}(k)+R(k+1)(z_k-\boldsymbol{v}(k)^{\mathrm{T}}\boldsymbol{y}_k)\boldsymbol{y}_k. \tag{3.5.23}$$

这种算法得到的权向量序列 $\{\boldsymbol{v}(k)\}$ 同样收敛于最优解向量 $\boldsymbol{v}^*$, 且收敛速度较快, 但迭代过程中每一步计算量较大.

3.6 多类问题

本章前几节讨论了二类问题的线性判别方法, 然而实际上经常遇到样本的多类分类问题. 因此必须研究多类问题的线性判别方法.

利用线性判别函数设计多类分类器有多种途径.

方法 (1)—— 把 c 类问题化为 c 个二类问题.

通过一个线性判别函数把一个类型的样本与其他类别区分开来, 对于 c 个类别, 需建立 c 个线性判别函数, 即

$$g_l(\boldsymbol{x})=\boldsymbol{w}_l^{\mathrm{T}}\boldsymbol{x}+w_{l0},\qquad l=1,2,\cdots,c \tag{3.6.1}$$

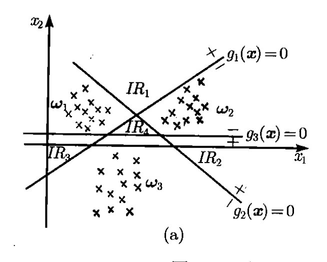
其中每一个判别函数有如下功能:

$$\begin{cases} g_l(\boldsymbol{x})>0, & 则\ \boldsymbol{x}\in\omega_l \\ g_l(\boldsymbol{x})\leqslant 0, & 则\ \boldsymbol{x}\notin\omega_l \end{cases}\qquad l=1,2,\cdots,c \tag{3.6.2}$$

判别规则为

$$\begin{cases} g_l(\boldsymbol{x})>0, \\ g_m(\boldsymbol{x})\leqslant 0,\quad m=1,2,\cdots,c,\quad m\neq l \end{cases}\qquad 则\ \boldsymbol{x}\in\omega_l. \tag{3.6.3}$$

这种方法的图示见图 3.9(a), 这时, 位于图中 IR_1,IR_2,IR_3,IR_4 区域的样本点分类器无法确定其类别. 原因是式 (3.6.3) 确定的 ω_l 可能会与 $\omega_m(m=1,2,\cdots,c,m\neq l)$ 相互重叠, 其重叠区域究竟属于 ω_l 还是 ω_m 无法判别; 还可能出现不属于任何类别的区域 (IR_4).



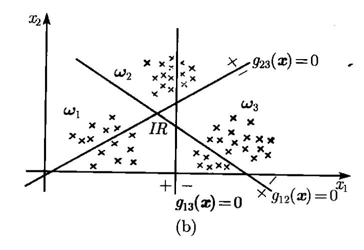


图 3.9 多类问题转化为多个二类问题的两种情况

方法 (2)—— 把 c 类问题化为 $c(c-1)/2$ 个二类问题.

对 c 个类型中任意两个类型 ω_l 和 ω_m 建立一个判别函数 $g_{lm}(\boldsymbol{x})$, 它将两个类型 ω_l 和 ω_m 区别开, 但对其他的类型不提供任何信息. 因为 c 个类型中任意两个类型 ω_l 和 ω_m 的组合数为 $c(c-1)/2$ 个, 所以共需建立 $c(c-1)/2$ 个判别函数, 即

$$g_{lm}(\boldsymbol{x}) = \boldsymbol{w}_{lm}^{\mathrm{T}}\boldsymbol{x} + w_{lm0}, \qquad l, m = 1, 2, \cdots, c, \quad l \neq m \tag{3.6.4}$$

它具有性质

$$g_{lm}(\boldsymbol{x}) = -g_{ml}(\boldsymbol{x}),$$

和如下功能:

$$\begin{cases} g_{lm}(\boldsymbol{x}) > 0, & \text{则 } \boldsymbol{x} \in \omega_{ml} \\ g_{lm}(\boldsymbol{x}) < 0, & \text{则 } \boldsymbol{x} \notin \omega_{ml} \end{cases} . \tag{3.6.5}$$

其判别规则为

$$\boldsymbol{x} \in \omega_l, \qquad \text{当 } g_{lm}(\boldsymbol{x}) > 0, m = 1, 2, \cdots, c, \quad l \neq m \tag{3.6.6}$$

即为了得到 $\boldsymbol{x} \in \omega_l$ 的结论, 必须考察 $c-1$ 个判别函数 $g_{lm}(\boldsymbol{x}), m = 1, 2, \cdots, c, l \neq m$. 这种方法的图示见图 3.9(b). 同样, 这时会有一个区域同时属于两个以上的类型, 即图中标记为 IR 的区域, 该区域的样本点分类器无法确定其类别.

方法 (3)—— 最大值判别规则

方法 (1) 和 (2) 的共同缺点是某一部分区域中的样本点无法分类. 这一缺点在方法 (3) 中得到了克服.

定义 c 个判别函数

$$g_l(\boldsymbol{x}) = \boldsymbol{w}_l^{\mathrm{T}}\boldsymbol{x} + w_{l0}, \qquad l = 1, 2, \cdots, c \tag{3.6.7}$$

判别规则为

$$g_l(\boldsymbol{x}) > g_m(\boldsymbol{x}), \quad m = 1, 2, \cdots, c, m \neq l, \qquad \text{则 } \boldsymbol{x} \in \omega_l. \tag{3.6.8}$$

这样的分类器称为线性机器, 它把特征空间分割为 c 个决策区域 $R_1, R_2, \cdots, R_c$, 样本 $\boldsymbol{x}$ 被归类为 $g_i(\boldsymbol{x})$ 在 c 个判别函数中取极大值的那个类别 l. 这种方法的优点在于不存在不确定区. 如果 R_l 与 R_m 相邻, 则它们的分界面就是超平面 H_{lm} 的一部分, 其定义为

$$g_l(\boldsymbol{x}) = g_m(\boldsymbol{x}) \tag{3.6.9}$$

或

$$(\boldsymbol{w}_l - \boldsymbol{w}_m)^{\mathrm{T}}\boldsymbol{x} + (w_{l0} - w_{m0}) = 0. \tag{3.6.10}$$

由此可知, $(\boldsymbol{w}_l-\boldsymbol{w}_m)$ 是 H_{lm} 的法向量, 从 $\boldsymbol{x}$ 到超平面 H_{lm} 的代数距离为

$$r=\frac{g_l(\boldsymbol{x})-g_m(\boldsymbol{x})}{\|\boldsymbol{w}_l-\boldsymbol{w}_m\|}, \tag{3.6.11}$$

因此, 对线性机器来说, 重要的是权向量的差而不是权向量本身. 这时, 应该有 $c(c-1)/2$ 个超平面, 但在实际问题中出现在分界面上的超平面的个数往往少于 $c(c-1)/2$ 个. 图 3.10 是在二维特征空间情况下的三类和五类问题线性决策面的示意图.

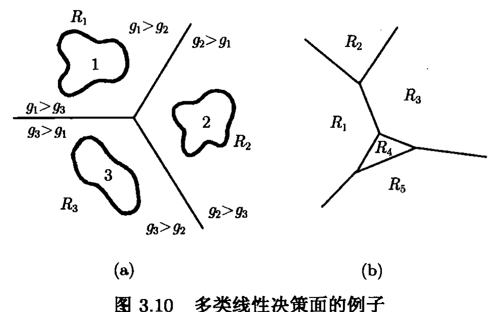


图 3.10 多类线性决策面的例子

(a) 三类; (b) 五类

第四章　决策树判别

前面我们讨论了用线性判别函数设计分类器的方法. 但是大量实际的模式识别问题并不是线性可分的, 比如当两类样本的分布具有多峰性质并相互交错时, 简单的线性判别函数往往会导致较大的分类错误. 这种情况下就需要采用非线性分类器.

从本章开始我们来讨论几种常用的、特别是在高能物理实验数据分析中常用的非线性分类方法.

4.1　超长方体分割法

我们首先讨论一种对于二类问题的最简单的非线性判别方法 —— 超长方体分割法, 它可以认为是决策树判别方法的一种最简单的特例, 但是由于它简单、易实行的特点, 在实验数据的多元分析中, 特别是高能物理实验数据分析中, 仍然有比较广泛的应用.

4.1.1　超长方体分割法的基本思想

在本节的讨论中, 为了不失一般性, 我们把样本分为信号和本底两个类别, 信号指实验中所要研究的过程的事例样本, 所有信号以外的样本都属于本底样本.

超长方体分割法不是企图用一个决策规则把两类样本一次分开, 而是采用分级的方法来解决分类问题. 它的基本思想如图 4.1 所示. 首先要根据分类问题的具体要求选择适当的特征向量 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$, 特征向量的每一个变量 x_j 都是实验的直接或间接测量值变量, 而且具有区分信号和本底的能力, 也就是说, 该变量的概率密度分布对于信号样本和本底样本有明显的差别, 能够用阈值 x_j^{th} 把变量域划分为两个区域: 类信号区和类本底区, 在类信号区中信号事例样本占有比较大的比例；类本底区中则本底事例样本占有比较大的比例. 把待分类样本集每个样本的特征向量的各个特征的测量值逐个输入分类器, 分类器按每个变量 x_j 的值将它归类到类信号区和类本底区, 如若第 j 个变量 x_j 被归入类信号区, 则再利用变量 x_{j+1} 的值对样本分类, 直到用变量 x_n 的值将样本归类为止. 一个所有变量值 $(x_j, j=1,2,\cdots,n)$ 都被归入类信号区的样本被分类器最终判别为信号样本, 其他所有样本被判为本底样本. 也就是说, 只要对任何一个变量的判别上被归入类本底区, 该样本就被分类器判别为本底事例样本.

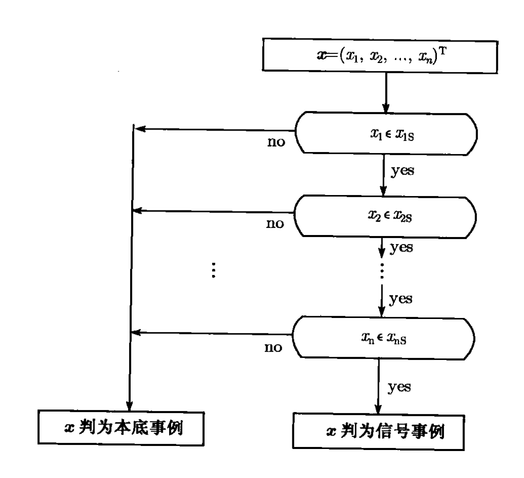


图 4.1 超长方体分割法区分二类样本的示意图

其中 $x_i \in x_{iS}$ 表示观测值 x_i 落入分类器规定的特征向量第 i 个变量的信号区内

假定信号事例样本集的样本总数为 $N_{\rm S}$, 经过分类器后被判为信号的样本数为 $n_{\rm SS}$, 则该分类器对于信号事例的选择效率为

$$\varepsilon_{\rm SS}=\frac{n_{\rm SS}}{N_{\rm S}}. \tag{4.1.1}$$

类似地, 若本底事例样本集的样本总数为 $N_{\rm B}$, 经过分类器后被判为信号的样本数为 $n_{\rm SB}$, 则该分类器的信号误判率为

$$\varepsilon_{\rm SB}=\frac{n_{\rm SB}}{N_{\rm B}}. \tag{4.1.2}$$

显然, 高的信号选择效率和低的误判率是我们追求的目标.

这种方法把每个变量 x_j 的值域看成是超长方体第 j 根轴的边长, 将每一根边长分割为类信号区和类本底区, 所以形象地称为超长方体分割法.

4.1.2 超长方体分割法中阈值的确定

从以上分类过程我们可以看到, 这种分类器设计最重要的问题是怎样将每个变量划分为类信号区和类本底区, 或者说, 怎样确定阈值向量 $\boldsymbol{x}^{\rm th}=(x_1^{\rm th},x_2^{\rm th},\cdots,x_n^{\rm th})^{\rm T}$. 对于一个待分类的实际的样本集, 如果有比较充分的先验知识, 能够构造一个分布相近的 "模拟" 样本集, 并且该样本集的样本数量足够大, 那么可以利用这个模拟

样本来确定阈值向量. 这种情况在粒子物理实验中具有典型性. 如研究正负电子对撞产生的末态 f:

$$\mathrm{e^+e^-} \to (\text{过程}1, 2, \cdots, k) \to f$$

其中过程 1 是我们感兴趣的信号, 其余过程均为本底. 对于所有这些过程产生的末态如果均有已知的理论模型以一定的精度加以描述, 并且各个过程产生末态 f 的相对强度亦为已知, 那么, 就可以用蒙特卡罗方法构造出一个与实际数据样本分布相近的 "模拟" 样本集, 并且该样本集的样本数量原则上可以无限地产生.

有了这样的模拟样本集, 可以得到每个变量的信号和本底样本的近似边沿概率密度

$$p_{j,\mathrm{mar}}^{\mathrm{S/B}} = p^{\mathrm{S/B}}\left(x_j | x_1, \cdots, x_{j-1}, x_{j+1}, \cdots, x_n \in (-\infty, +\infty)\right). \tag{4.1.3}$$

式中, 上标 S 表示信号, B 表示本底. 利用信号和本底样本的 x_j 的边沿概率密度的差别, 容易确定阈值 x_j^{th} 值. 如果信号和本底样本的 x_j 的边沿概率密度是分离的, 如图 4.2 所示, 那么 x_j^{th} 可以取为分离区内的任意 x_j 值. 这时判选规则为 $x_j > x_j^{\mathrm{th}}$ 归入类信号区, 否则归入类本底区. 该判据对于信号事例的选择效率为 $\varepsilon_{\mathrm{SS}} = 1$, 将本底事例误判为信号事例的误判率为 $\varepsilon_{\mathrm{SB}} = 0$.

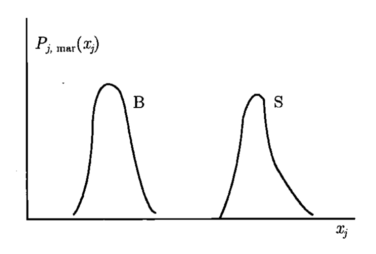


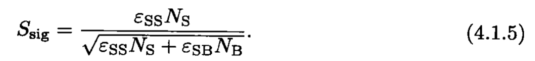
图 4.2　阈值 x_j^{th} 的确定: $p_{j,\mathrm{mar}}^{\mathrm{S}}$ 与 $p_{j,\mathrm{mar}}^{\mathrm{B}}$ 分离的情形

但是, 一般情形下 $p_{j,\mathrm{mar}}^{\mathrm{S}}$ 与 $p_{j,\mathrm{mar}}^{\mathrm{B}}$ 是相互重叠而不相分离的, 如图 4.3(a) 所示. 这种情形下, 粒子物理实验中往往要求 "信号事例" 的判选规则使得对于事例的选择具有最大的 "信号显著性". 信号显著性定义为

$$S_{\mathrm{sig}} = \frac{n_{\mathrm{SS}}}{\sqrt{n_{\mathrm{SS}} + n_{\mathrm{SB}}}}, \tag{4.1.4}$$

式中, $n_{\mathrm{SS}}, n_{\mathrm{SB}}$ 分别为经过该判选规则后类信号区内的信号和本底事例数. 信号显著性越高, 类信号区内的信号越清晰. 假定模拟样本集在施加该判选规则前的信号和本底事例数分别为 N_{S} 和 N_{B}, 该判选规则对于信号和本底事例的 "信号" 选择

效率为 $\varepsilon_{\rm SS}$ 和 $\varepsilon_{\rm SB}$, 当总事例数 $N=N_{\rm S}+N_{\rm B}$ 足够大时, 信号显著性可用下式确定:



$$S_{\rm sig}=\frac{\varepsilon_{\rm SS}N_{\rm S}}{\sqrt{\varepsilon_{\rm SS}N_{\rm S}+\varepsilon_{\rm SB}N_{\rm B}}}. \tag{4.1.5}$$

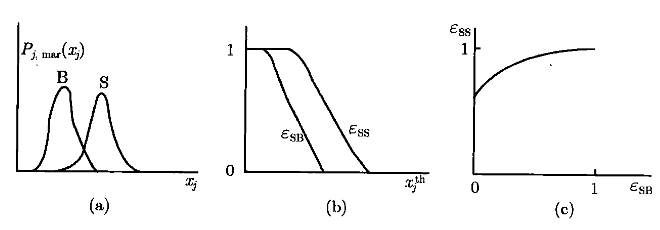


图 4.3 阈值 $x_j^{\rm th}$ 的确定: $p_{j,\rm mar}^{\rm S}$ 与 $p_{j,\rm mar}^{\rm B}$ 重叠的情形

(a) 信号和本底样本的边沿概率密度 $p_{j,\rm mar}^{\rm S}$ 与 $p_{j,\rm mar}^{\rm B}$; (b) 信号和本底事例的 "信号" 选择效率 $\varepsilon_{\rm SS}$ 和 $\varepsilon_{\rm SB}$ 与阈值 $x_j^{\rm th}$ 间的关系曲线; (c) $\varepsilon_{\rm SS}$ 与 $\varepsilon_{\rm SB}$ 间的关系曲线

将 $\varepsilon_{\rm SB}$ 视为 $\varepsilon_{\rm SS}$ 的函数, $S_{\rm sig}$ 的极大值可由求方程 $\dfrac{{\rm d}S_{\rm sig}}{{\rm d}\varepsilon_{\rm SS}}=0$ 的根得到, 其解为

$$\varepsilon_{\rm SB}=\frac{\varepsilon_{\rm SS}}{2N_{\rm B}}\left(N_{\rm B}\frac{{\rm d}\varepsilon_{\rm SB}}{{\rm d}\varepsilon_{\rm SS}}-N_{\rm S}\right). \tag{4.1.6}$$

由图 4.3(a) 的信号和本底的条件概率密度 $p_{j,\rm mar}^{\rm S}$ 与 $p_{j,\rm mar}^{\rm B}$, 容易求得信号和本底事例的选择效率 $\varepsilon_{\rm SS}$ 和 $\varepsilon_{\rm SB}$ 与阈值 $x_j^{\rm th}$ 间的函数关系, 如图 4.3(b) 所示, 进一步可得到 $\varepsilon_{\rm SS}$ 与 $\varepsilon_{\rm SB}$ 间的函数关系, 如图 4.3(c) 所示, 并可求得 $\dfrac{{\rm d}\varepsilon_{\rm SB}}{{\rm d}\varepsilon_{\rm SS}}$ 与 $\varepsilon_{\rm SS}$ 的函数关系. 这样从图 4.3(c) 的曲线就能找出满足式 (4.1.6) 的 $\varepsilon_{\rm SS}$ 和 $\varepsilon_{\rm SB}$ 值, 再由图 4.3(b) 求得 $S_{\rm sig}$ 的极大值对应的阈值 $x_j^{\rm th}$.

也可考虑求量 $H_{\rm sig}=\varepsilon_{\rm SS}S_{\rm sig}$ 的极大值来确定阈值 $x_j^{\rm th}$. 这意味着判选规则要求信号选择效率和显著性的乘积达到极大, 即阈值 $x_j^{\rm th}$ 的选择不但考虑到有尽可能高的信号显著性, 还考虑到有尽可能高的信号选择效率; 因为只有高的信号显著性而信号选择效率很低, 并不是一个好的事例分类器. 求方程 $\dfrac{{\rm d}H_{\rm sig}}{{\rm d}\varepsilon_{\rm SS}}=0$ 的根, 其解为

$$\varepsilon_{\rm SB}=\frac{\varepsilon_{\rm SS}}{4N_{\rm B}}\left(N_{\rm B}\frac{{\rm d}\varepsilon_{\rm SB}}{{\rm d}\varepsilon_{\rm SS}}-3N_{\rm S}\right). \tag{4.1.7}$$

$H_{\rm sig}$ 的极大值对应的阈值 $x_j^{\rm th}$ 可用与前述步骤类似的方法和式 (4.1.7) 得到.

4.1.3 超长方体分割法的优缺点及其改进

超长方体分割法的显著优点是设计分类器十分简单. 从以上分类过程我们可以看到, 这种分类器设计最重要的问题是怎样将每个变量划分类信号区和类本底

区, 或者说, 怎样确定阈值向量 $\boldsymbol{x}^{\mathrm{th}}=(x_1^{\mathrm{th}},x_2^{\mathrm{th}},\cdots,x_n^{\mathrm{th}})^{\mathrm{T}}$. 只要有了足够数量的信号和本底事例的模拟样本集, 可以由 4.1.2 小节所述的方法确定 x_j^{th} 值. 其次, 分类器的设计有很大的灵活性, 只要 x_j^{th} 值相同, x_j 出现的先后次序不同并不改变分类器的信号选择效率 $\varepsilon_{\mathrm{SS}}$.

但超长方体分割法的缺陷也很明显. 首先, 它实际上是一系列的单变量分析, 没有用到多个变量的组合信息, 所以还不是真正意义上的多元变量分析方法. 只要对任何一个变量的判别上被归入类本底区, 该样本就被归类为本底, 实际上被归入类本底区的样本其中有一部分是信号样本, 只不过需要通过其他几个变量或变量组合才能判别出来. 所以超长方体分割法不可能将需要变量组合才能判别出来的信号样本判定为信号, 这就使得它的信号选择效率 $\varepsilon_{\mathrm{SS}}$ 往往是比较低的. 对于待分类样本集中信号样本数比例本来就比较小的情况, 这一缺陷尤其明显.

第二个缺点是对于选定的特征向量 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$, 要确定一组最优的阈值向量 $\boldsymbol{x}^{\mathrm{th}}=(x_1^{\mathrm{th}},x_2^{\mathrm{th}},\cdots,x_n^{\mathrm{th}})^{\mathrm{T}}$ 十分困难. 一个好的超长方体分割法分类器, 其基本原则是一个待分类的样本集经过该分类器后, 在最后一个变量 x_n 的类信号区内的样本数 n_t(即分类器判别为信号事例的样本数) 中, 能包含尽可能多的信号事例样本, 而本底事例样本数尽可能地少, 使得在这一区域内信号事例数对于本底事例数有比较高的信号显著性. 但是对于选定的特征向量 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$, 要确定一组最优的阈值向量 $\boldsymbol{x}^{\mathrm{th}}=(x_1^{\mathrm{th}},x_2^{\mathrm{th}},\cdots,x_n^{\mathrm{th}})^{\mathrm{T}}$ 使得信号显著性达到极大却十分困难. 原因在于在利用信号和本底事例的训练样本确定 x_j 的阈值 x_j^{th} 时, 方便的做法是利用信号和本底样本的式 (4.1.3) 所示的 x_j 的边沿概率密度 $p_{j,\mathrm{mar}}$ 的差别来确定最佳 x_j^{th} 值 (如使得由该 x_j^{th} 值确定的类信号区内的信号显著性达到极大). 但在超长方体分割法中, 应该用信号和本底样本的 x_j 的条件概率密度

$$p_{j,\mathrm{con}}^{\mathrm{S/B}}=p^{\mathrm{S/B}}\left(x_j|x_1\in S_1,\cdots,x_{j-1}\in S_{j-1};x_{j+1},\cdots,x_n\in(-\infty,+\infty)\right)\qquad(4.1.8)$$

的差别来确定最佳 x_j^{th} 值, 式中 $x_k\in S_k$ 表示 x_k 被归入类信号区. 所以用 $p_{j,\mathrm{mar}}$ 确定的最佳阈值向量并不是超长方体分割法的真正的最佳阈值向量 $\boldsymbol{x}^{\mathrm{th}}$.

第三个缺点是特征向量 $\boldsymbol{x}$ 的 n 个变量是否都需要用来作分类判别往往是不明确的, 可能其中的一些变量对于不同样本的分辨能力已被其他变量所覆盖因而是不必要的.

超长方体分割法的以上缺点可以有以下途径进行改进.

(1) 利用条件概率密度确定阈值向量 $\boldsymbol{x}^{\mathrm{th}}$

如果已有足够数量的信号和本底事例的模拟样本集, 那么可以用来构造式 (4.1.8) 所示的信号和本底样本的 x_j 的条件概率密度 $p_{j,\mathrm{con}}^{\mathrm{S}}$ 和 $p_{j,\mathrm{con}}^{\mathrm{B}}$, 利用条件概率密度 $p_{j,\mathrm{con}}^{\mathrm{S}}$ 和 $p_{j,\mathrm{con}}^{\mathrm{B}}$ 的差别, 来确定阈值 x_j^{th} 值. 在模拟样本集与实际数据样本集

分布相近的条件下, 这样确定的阈值接近最佳阈值向量 $\boldsymbol{x}^{\text{th}}$. 但在实际操作中, 逐级计算条件概率密度是相当麻烦的一件事.

(2) 在某些节点对若干个特征变量用线性判别函数进行判别

超长方体分割法的基本框架中, 每个节点仅用一个特征变量进行判别. 但这不是强制性的要求. 完全可以在任何一个节点对于已知存在线性关联的若干个特征变量利用第三章所述的任何一种线性判别方法进行判别. 这在超长方体分割法的架构中非常容易实现, 并且能有效地提高信号判选效率, 降低误判率.

(3) 对样本数据 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\text{T}}$ 首先进行主成分分析

如果对样本数据 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\text{T}}$ 首先进行主成分分析得到新特征向量数据 $\boldsymbol{y}=(y_1,y_2,\cdots,y_n)^{\text{T}}$, 然后用超长方体分割法对新特征向量 $\boldsymbol{y}$ 进行信号和本底样本的分类, 则上面所述的这些缺陷在一定程度上能得到克服. 首先它利用了原特征线性组合的信息, 有效地提高了信号的选择效率. 其次, 由于各个 y_j 之间的线性相关系数为 0, 利用信号和本底事例的训练样本得到每个变量的近似边沿概率密度

$$p_{j,\text{mar}}=p\left(y_j|y_1,\cdots,y_{j-1},y_{j+1},\cdots,y_n\in(-\infty,+\infty)\right). \tag{4.1.9}$$

来确定阈值向量 $\boldsymbol{y}^{\text{th}}=(y_1^{\text{th}},y_2^{\text{th}},\cdots,y_n^{\text{th}})^{\text{T}}$ 比较接近于最优的阈值向量. 最后, 如果最后几个主成分的方差贡献率足够小, 则我们可以作降维处理, 既减小了计算量, 又不降低分类器对信号样本的判别能力. 对样本数据 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\text{T}}$ 进行主成分分析并不需要各特征变量之间是否线性相关的知识, 因此在实际应用中比第二种方法更易于实现.

4.1.4 超长方体分割法用于高能物理实验分析

作为一个例子, 我们简要地说明超长方体分割法用于研究 e^+e^- 对撞中 $\psi(2S)\to\text{p}\bar{\text{p}}$ 反应中怎样从大量本底事例中判选出信号事例 [18]. 北京谱仪国际合作研究组 (BES Collaboration) 利用 e^+e^- 对撞在质心系能量 $E_{\text{cm}}=3.686\text{GeV}$ 处产生了 14 兆 $\psi(2\text{S})$ 粒子, 由于它衰变为 $\text{p}\bar{\text{p}}$ 信号事例的分支比 (即概率) 仅为 $(3.36\pm0.27)\times10^{-4}$, 可见排除本底的要求十分高. 这种情况在粒子物理实验中是相当典型的. 显而易见, 如果要研究分支比更低的衰变过程, 排除本底的要求必定更为苛刻.

首先要选择适当的特征向量 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\text{T}}$, 特征向量的每一个变量 x_j 都是实验的直接或间接测量值变量, 而且具有区分信号和本底的能力. 根据 $\psi(2\text{S})\to\text{p}\bar{\text{p}}$ 反应与其他本底事例的不同特性, 我们用来选择信号事例的事例判选规则用到了以下的变量.

1. 带电径迹数条件 N_C=2

带电径迹数条件 N_C 是北京谱仪子探测器主漂移室确定的特征量之一, 它表示主漂移室测到的带电径迹条数. 信号事例 $\psi(2\text{S})\to\text{p}\bar{\text{p}}$ 末态只有 $\text{p}\bar{\text{p}}$ 两个带电粒

子, 这一判选条件排除了所有末态带电粒子数不等于 2 的大量本底. 该条件物理上不造成信号事例的效率损失, 但主漂移室的有效探测立体角约为 4π 立体角的 85%, 因此存在探测器的有效探测立体角导致的信号事例的效率损失. 凡满足本级判选条件的事例归入本级判选中的类信号事例 (下同, 不再重复).

2. 径迹飞行时间条件

对于每根径迹要求

$$|t_{\rm m}-t_{\rm p}|<|t_{\rm m}-t_{\rm K}|,|t_{\rm m}-t_{\pi}|,|t_{\rm m}-t_{\mu}|,|t_{\rm m}-t_{\rm e}|,$$

式中, $t_{\rm m}$ 是子探测器飞行时间计数器 (TOF) 测到的实际飞行时间; $t_i, i={\rm p},{\rm K},\pi,\mu,{\rm e}$ 是假设径迹是粒子 i, 根据粒子的能量 (等于质心系能量的一半 1.843GeV), 对撞中心到 TOF 系统的飞行长度, 以及粒子 i 的质量计算出来的飞行时间. 显然, 如果是信号事例, $|t_{\rm m}-t_{\rm p}|$ 应该接近于 0, 且比其他几个时间差值要小. 因此该判选条件能够排除大量 $e^+e^-,\mu^+\mu^-,\pi^+\pi^-,K^+K^-$ 两体末态的本底事例, 而对信号事例的判选效率不造成物理上的损失. 但由于飞行时间计数器的有效探测立体角为 4π 的 76%, 因此存在探测器的有效探测立体角导致的信号事例的效率损失. $t_i, i={\rm p},{\rm K},\pi,\mu,{\rm e}$ 的计算中需要用到径迹从对撞中心到击中 TOF 的飞行长度, 它是根据该径迹在主漂移室中的飞行轨迹推算出来的, 因此该判选条件实际上用到了两个特征变量, 即飞行时间和飞行长度.

3. 两径迹飞行时间差条件 $\Delta t<4{\rm ns}$

该条件用以排除宇宙线本底. $\Delta t=|t_+-t_-|$ 表示 TOF 计数器测到的两根径迹的飞行时间之差, 所以该条件用到了 t_+,t_- 两个特征量. 对于 e^+e^- 对撞产生的两个动量相等的带电粒子以相反方向飞出的事例, $\Delta t=0$; 对于穿过对撞中心的宇宙线事例, $\Delta t=8{\rm ns}$. 由于测量误差, 实际的 Δt 是以 0 和 $\pm 8{\rm ns}$为中心值的分布, 如图 4.4 所示. 该条件几乎能排除所有的宇宙线本底, 对于信号事例的选择效率几乎没有物理的损失,

4. 径迹背对背条件 $\theta<5^\circ$

信号事例 $\psi(2S)\to p\bar{p}$ 末态 $p\bar{p}$ 两个带电粒子以相反方向飞出, 因此物理上两条带电径迹间的夹角 θ 应为 0. θ 需要从两条径迹的方向参数求出, 所以该条件用到了两组特征参数. 图 4.5 是 $\psi(2S)\to p\bar{p}$ 信号的蒙特卡罗模拟训练样本和本底的蒙特卡罗模拟训练样本 (信号事例数与本底事例数已经正确地归一化了, 即与真实数据中的比例一致) 的 θ 分布图. $\psi(2S)\to p\bar{p}$ 的 θ 大于 0 是由于带电径迹探测和重建导致的径迹方向误差以及粒子的多次库仑散射造成的方向误差. 区分信号和本底的阈值选为 5°. 该条件对信号事例的判选效率损失很小, 但排除了大量不满足背对背条件的两径迹末态本底事例.

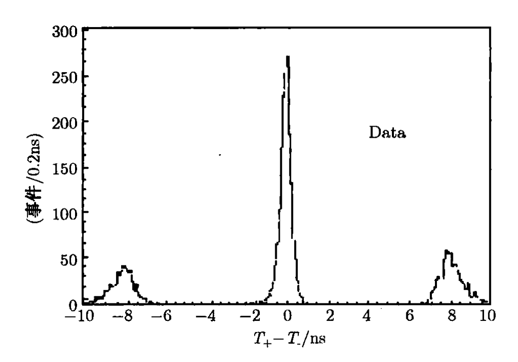


图 4.4 两带电径迹飞行时间差分布

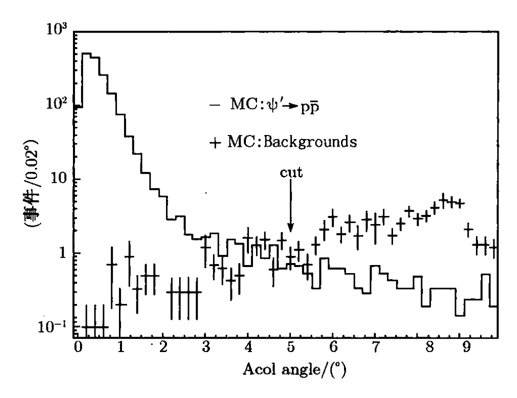


图 4.5 背对背径迹飞行方向夹角的分布

直方图代表 $\psi(2S) \to p\bar{p}$ 信号的蒙特卡罗模拟训练样本, 十字叉代表本底的蒙特卡罗模拟训练样本. 信号事例数与本底事例数已经正确地归一化. 注意 y 轴是对数坐标

经过以上判选条件, 实验数据样本中除信号事例外, 只余下少量下述的背对背两径迹本底实例:

$$e^+e^- \to e^+e^-, \mu^+\mu^-, \pi^+\pi^-, K^+K^-,$$

$$\psi(2S) \to e^+e^-, \mu^+\mu^-, \pi^+\pi^-, K^+K^-, \tag{4.1.10}$$

5. 带正电粒子沉积能量条件 $E_+ < 0.75\text{GeV}$

子探测器电磁量能器测量粒子在其中的沉积能量. 在北京谱仪的情况下, 测得

的正负电子的沉积能量的中心值与其实际能量相近, 而测得的质子的沉积能量的中心值比它的实际能量小得多. 图 4.6 所示为带正电粒子沉积能量的分布, 其中直方图代表 $\psi(2S) \to p\bar{p}$ 信号的蒙特卡罗模拟训练样本中的 p 的沉积能量, 集中于低能端. 十字叉代表实验数据经过事例判选条件后选出的事例中带正电粒子的沉积能量, 其中低能端的分布来自于 $\psi(2S) \to p\bar{p}$ 信号事例的贡献, 所以与蒙特卡罗模拟的结果十分接近, 高能端的突起来自于本底事例 $e^+e^- \to e^+e^-, \psi(2S) \to e^+e^-$ 中的 e^+ 的沉积能量. e^+ 的实际能量等于或接近质心系能量 $E_{\rm cm}$ 的一半 1.843GeV, 由于电磁量能器的能量分辨较差 $(\Delta E/E = 0.22/\sqrt{E(\rm GeV)})$, 所以形成以 1.843GeV为中心的一个宽的分布. 判选条件 $E_+ <$0.75GeV 可以排除 $e^+e^- \to e^+e^-$ 和 $\psi(2S) \to e^+e^-$ 导致的本底, 而对信号事例的判选效率损失很小.

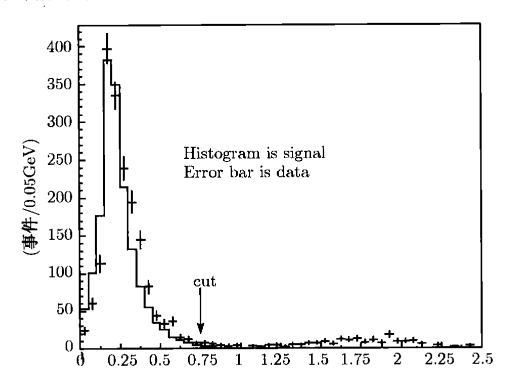


图 4.6　带正电粒子沉积能量的分布, 横轴单位为 GeV

直方图代表 $\psi(2S) \to p\bar{p}$ 信号的蒙特卡罗模拟训练样本经过事例判选条件选出的事例中的 p 的沉积能量, 十字叉代表实验数据经过事例判选条件选出的事例中带正电粒子的沉积能量. 信号事例数与本底事例数已经正确地归一化

6. $p\bar{p}$ 总能量判选条件

用 $E_{\rm p}$ 表示用带正电粒子动量和质子质量计算得到的能量值, $E_{\bar{\rm p}}$ 表示用带负电粒子动量和反质子质量计算得到的能量值. 对于信号事例 $\psi(2S) \to p\bar{p}$, $E_{\rm p} + E_{\bar{\rm p}}$ 应该与 $\psi(2S)$ 的质量值 (3.686GeV) 一致. 考虑到动量测量存在误差, 特征量 $|E_{\rm p} + E_{\bar{\rm p}} - 3.686|$ 应当与 0 相差不大. 而对于式 (4.1.6) 所示的本底事例, 因为用错误的粒子质量 (质子质量) 计算 $E_{\rm p}$ 和 $E_{\bar{\rm p}}$, 其能量和 $E_{\rm p} + E_{\bar{\rm p}}$ 与 $\psi(2S)$ 的质量值差别比较大. 图 4.7 表示了 $E_{\rm p} + E_{\bar{\rm p}}$ 的分布. 其中直方图代表 $\psi(2S) \to p\bar{p}$ 信号的蒙特卡罗模拟训练样本的分布, 阴影部分表示本底的蒙特卡罗模拟训练样本的分布.

十字叉代表实验数据的分布, 所有上述事例样本都经过事例判选条件选出. 可以看到实验数据的分布与信号加本底的蒙特卡罗模拟训练样本的分布比较接近, 但在高能量端实验数据的事例数比较多, 说明对于本底的蒙特卡罗模拟还有缺陷. 采用 $|E_{\rm p}+E_{\bar{\rm p}}-3.686|<0.13$GeV 的判选条件使得对于信号事例有较高的效率, 并且避免了蒙特卡罗模拟的缺陷带来的不一致性. 该判选条件用到了两个特征量.

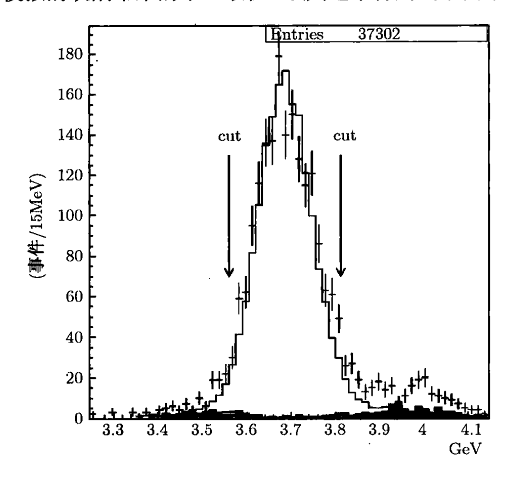


图 4.7 $E_{\rm p}+E_{\bar{\rm p}}$ 的分布

直方图代表 ψ(2S) → p$\bar{\rm p}$ 信号的蒙特卡罗模拟训练样本的分布, 阴影部分表示本底的蒙特卡罗模拟训练样本事例的分布. 十字叉代表实验数据事例的分布. 所有上述事例样本都经过事例判选条件选出, 信号事例数与本底事例数已经正确地归一化

7. 带负电粒子动量判选条件

对于 ψ(2S) → p$\bar{\rm p}$ 信号事例, 带负电粒子 (反质子) 动量 $p_{\bar{\rm p}}$ 应为 1.586GeV. 而对于式 (4.1.10) 所示的本底事例, 因为 e, μ, π, K 的质量远远小于质子质量, 所以粒子动量高于 1.586GeV, 分别为 1.843, 1.840, 1.838, 1.775GeV. 所以用判选条件 $|p_{\bar{\rm p}}-1.586|<0.15$GeV 可以将信号和本底区分开来, 有高的信号判选效率和强的本底排除能力. 之所以有 0.15GeV的宽容是考虑到反质子在主漂移室中的能量损失和动量测定的不确定性.

实验数据样本经过以上事例判选条件后, 或者说经过上述判选条件构成的信号事例分类器, 得到的类信号事例的带正电粒子的动量分布如图 4.8 中数据点所示. 图中的直方图表示的是归一化的信号和本底的蒙特卡罗模拟训练样本的分布, 与实验数据的分布十分符合. 阴影部分是本底的蒙特卡罗模拟训练样本的分布, 它的事

例数占所选出的全部事例的比例很小，这一比例就是信号事例分类器的错误率，或者用粒子物理的语言称为本底事例污染率. 该信号事例分类器对于信号事例的选择效率为 $\varepsilon_{\rm SS}$ =34.4%, 考虑到一部分效率是由于探测器的有限立体角损失掉的，这一信号选择效率是相当高的. 如果把探测器的有限立体角损失考虑进去，实际的信号选择效率达到 $\varepsilon'_{\rm SS}$ =77.6%. 所选出的全部类信号事例数为 1656, 其中真实信号事例数 1618, 本底事例数为 38, 所以本底污染率为 2.29%. 该信号事例分类器对于本底事例的选择效率仅为 $\varepsilon_{\rm SB} = 38/(14\times10^6) = 2.7\times10^{-6}$, 所以对于本底事例有很强的排除能力.

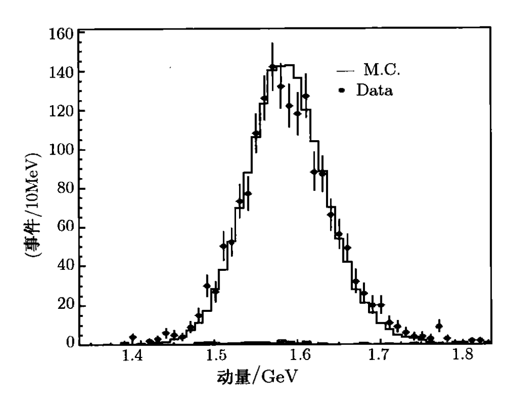


图 4.8　带正电粒子的动量分布

数据点代表实验数据事例的分布，直方图代表 $\psi(2S)\to p\bar{p}$ 信号和本底的蒙特卡罗模拟训练样本的分布，阴影部分表示本底的蒙特卡罗模拟训练样本事例的分布. 所有上述事例样本都经过事例判选条件选出，信号事例数与本底事例数已经正确地归一化

从这一具体实例我们可以看到超长方体分割法用于分类问题的一些特点. 首先，每一级判选中用到的判别量往往有明确的物理含义. 我们对于信号和本底关于该变量的分布往往已经有先验知识，知道这两者存在差别，因而可以利用它鉴别信号和本底. 其次，每一级判选中的判别量可以是一个变量，或者是若干个变量的某种组合或函数，后者对于信号和本底的鉴别有更强的能力，这种做法实际上是对超长方体分割法的一种简单而有效的改进，有助于提高信号选择效率，压制本底污染率. 第三，每一级判选中判别量阈值的确定可以利用判别量的分布直观地加以确定. 如果信号和本底训练样本的分布相互重叠，可用式 (4.1.3) 所示的信号显著性极大化加以确定. 而且信号和本底训练样本的分布相互离散的程度也反映了该判别量对于信号和本底判别能力的强弱. 再者，对于分类器最终选出的类信号事例，并不

简单地都认定为信号事例, 而是利用训练样本确定其污染率后把其中的本底污染事例数加以扣除以进一步减小测量误差. 由于以上这些做法, 超长方体分割法用于分类问题时, 不但具有简便, 物理图像明确的优点, 有时也能达到相当高的信号选择效率和本底排除能力.

4.2 决策树法

4.2.1 决策树法的基本思想

决策树 (decision trees) 或称树分类器[19,20], 是模式识别中进行分类的一种有效方法. 它在高能物理中的应用见文献 [21]. 利用树分类器可以把一个复杂的多类别分类问题转化为若干个简单的分类问题来解决. 它不是企图用一个决策规则把多个类别的样本一次分开, 而是采用分级的方法, 使分类问题逐步得到解决. 图 4.9 就是一个决策树的例子.

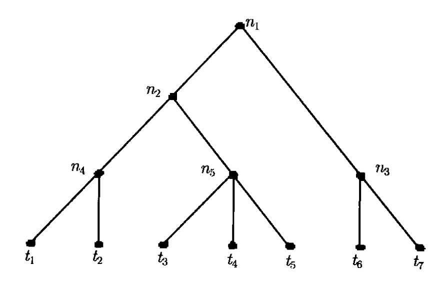


图 4.9 决策树示意图

一般地, 一个决策树由一个根节点 n_1, 一组非终止节点 n_i, 和一些终止节点 (称为叶节点) t_j 构成. 每个叶节点标以相应的样本类别标签, 不同的叶节点可以有相同的类别标签. 如果用符号 T 表示决策树, 那么一个 T 决策树对应于特征空间的一种划分, 它把特征空间分成若干个区域, 其中某个类别的样本占有优势的区域标记以该类样本的类别.

决策树的一种简单形式是二叉树. 所谓二叉树, 是指除叶节点外, 树的每个节点仅分为两个分支, 也就是说, 每个节点有且仅有两个子节点. 二叉树结构的分类器可以把一个复杂的多类别分类问题化为多级、多个两类问题来解决, 在每个节点都把样本集分为左、右两个子集. 分出的每个部分仍然可能包含多个类别的样本, 在下一级的节点, 把每个部分再分成两个子集, …… 直到最后分出的每个部分只包含同一类别的样本, 或某一类别样本占优势为止.

这种二叉树结构分类器概念简单、直观, 便于解释, 而且在各个节点上可以选择不同的特征和采用不同的决策规则, 因此设计方法简便且灵活多样, 便于利用先验知识来设计分类器.

图 4.10 是一个二叉决策树的例子. 该例中每个节点只选择一个特征, 并给出了相应的决策阈值. 对于未知样本 $\boldsymbol{x}$, 只要从根节点到叶节点顺序把 $\boldsymbol{x}$ 的某个特征观测值与相应的决策阈值比较, 就可作出决策, 把样本 $\boldsymbol{x}$ 分到相应的分支, 最后分到合适的类别.

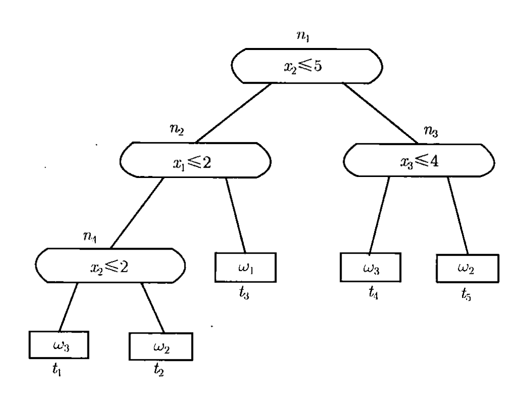


图 4.10　一个二叉决策树的示意图

从图 4.1 和 4.10 的对比可以看到, 超长方体分割法是二叉决策树方法的一种最简单的特例. 即使对于样本只分成信号和本底两个类别的两类问题, 并且每个节点仅对一个特征变量作二元决策的情形, 超长方体分割法和二叉决策树方法虽然是相似的, 却并不完全相同. 前者在每个节点的判别中必定将一部分样本判定为该分类器的 "本底事例", 后者却没有这样的要求; 前者一般是有 n 层节点, 顺次地利用特征向量 $\boldsymbol{x}$ 的 n 个变量进行判别, 每个变量利用一次 ; 而二叉决策树在每一个节点, 是通过某种优化步骤, 寻找特征向量 $\boldsymbol{x}$ 的某一个变量及其阈值, 使得在这一节点的判选中能最有效地区分信号和本底, 也就是说, 每个节点的判别变量是选择该节点中区分信号和本底能力最强的那个变量, 所以在一个二叉决策树中, 同一个变量可能在不同层次的节点中被重复使用.

设计一个决策树, 主要应解决三个问题:

1. 选择一个合适的树结构, 即合理安排树的节点和分支;
2. 确定每个非终止节点上要使用的特征;
3. 在每个非终止节点上选择适当的决策规则.

这三个问题解决了, 决策树的设计也就完成了. 二叉决策树的设计也不例外. 对于超长方体分割法, 也就是只有两种类别的最简单的二叉树, 其结构尤其简单, 其一般形式如图 4.1 所示.

把一个多类别问题转化为多个两类问题的途径是多种多样的, 因此, 对应的二叉树的结构也将各不相同. 因此决策树设计的目标是要寻找一个性能最优的决策树. 显然, 一个性能良好的决策树应该有高的判别效率和低的误判率, 以及尽可能小的计算量. 但是, 由于很难把效率和误判率的表达式与树的结构联系起来; 同时, 在每个节点上的决策规则也仅仅是该节点上所使用的特征观测值的函数, 即使每个节点上采用的决策规则性能达到最优, 由于没有考虑到与其他特征的可能的关联, 也不能说整个决策树的性能达到最优. 所以性能最优的决策树是很难达到和准确判断的. 在实际问题中, 人们往往提出其他一些优化准则, 例如极小化整个决策树的节点数, 或极小化从根节点到叶节点的最大路程长度, 或极小化从根节点到叶节点的平均路程长度等等, 力争设计出性能比较优良的决策树. 此外, 我们在超长方体分割法中讨论的优化决策规则性能的三种途径, 显然在决策树的优化设计中同样适用.

4.2.2 信号/本底二元决策树的构建

现在我们来讨论实验数据分析中常见的情形, 即解决信号和本底的两类事例的分类问题. 求解这类问题的过程, 就是利用一个训练样本集来构建 (训练) 一个决策树的过程. 训练样本集中包含信号和本底两类事例. 训练从根节点开始, 在每一个节点, 通过某种优化步骤, 寻找特征向量 $\boldsymbol{x}$ 的某一个变量及其阈值, 使得在这一节点的判选中能最有效地区分信号和本底. 通过该节点的判选, 输入事例被区分为 "类信号事例" 和 "类本底事例" 两部分, 其中 "类信号事例" 中信号事例的比率高于判选前的信号事例的比率, 而 "类本底事例" 部分则相反. 这两部分事例作为下一层节点的输入进行进一步的判选. 这一过程一直延续下去, 直到满足某种终结条件时停止. 最底层的 (叶) 节点被分为信号和本底节点两类, 其中到达信号事例多的叶节点指定为信号节点, 到达本底事例多的叶节点指定为本底节点. 这样, 一个决策树就构造完成了. 当一个待分类的样本集输入决策树, 则落入信号叶节点的事例被判定为 "信号事例", 落入本底叶节点的事例被判定为 "本底事例". 图 4.11 是一个区分信号/本底的二叉决策树的示意图.

问题在于怎样来评价和确定每个节点选择的 (变量 + 阈值) 组合对于信号和本底的判别能力. 事实上, 对于同样的 (变量 + 阈值) 组合, 它对信号和本底的判别能力取决于输入该节点的 (信号/本底) 事例数之比 r. 当 $r = 0$ 或 $r = \infty$(输入该节点的只有信号事例或只有本底事例), 任何 (变量 + 阈值) 组合都失去了对信号和本底的判别能力; 而当 $r = 1$ 时, 任何 (变量 + 阈值) 组合都达到其可能有的

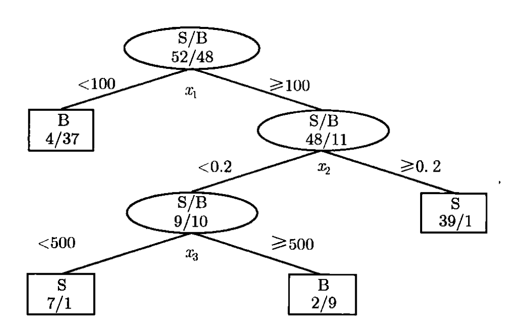


图 4.11　一个区分信号/本底的二叉决策树的示意图

图中方框表示叶节点, S 标志信号节点, B 标志本底节点. 所有节点中的左侧数字表示输入该节点的信号事例数, 右侧数字表示本底事例数

最大信号和本底的判别能力. 这种现象称为信号/本底判别能力对于输入样本成分的不均衡 (disparity) . 因此在每一节点选择最优 (变量 + 阈值) 组合时必须避免这种不均衡.

测试表明, 利用下列量来估价信号/本底判别能力不存在明显的性能不均衡:

Gini 指数 (Gini index) —— 定义为 $p(1-p)$ (4.2.1)

交叉熵 (cross entropy) —— 定义为 $-p\ln p-(1-p)\ln(1-p)$ (4.2.2)

误判误差 —— 定义为 $1-\max(p,1-p)$ (4.2.3)

统计显著性 —— 定义为 $n_S/\sqrt{n_S+n_B}$ (4.2.4)

其中, n_S, n_B 分别为输入该节点的信号和本底的事例数; p 表示该节点中输入的信号事例所占的比例, 也称为信号事例纯度 (purity):

$$p=n_S/(n_S+n_B) \tag{4.2.5}$$

式 (4.2.1)~(4.2.4) 所示的这几个量被称为 (信号/本底) 判别指数 (separation index), 用符号 I 表示. 决策树的训练过程中, 在每一个节点处在所有 n 个变量中只选择一组 (变量 + 阈值) 组合, 使得该节点的判别指数与它的两个子节点的判别指数的加权和的增量达到最大, 子节点的权值等于子节点的输入事例数除以母节点的输入事例数, 该增量 ΔI 用公式表示为

$$\Delta I=I-\left(\frac{n_1}{n_{\text{int}}}I_1+\frac{n_2}{n_{\text{int}}}I_2\right),\qquad n_{\text{int}}=n_1+n_2. \tag{4.2.6}$$

式中, I, I_1, I_2 分别为母节点和两个子节点的判别指数; n_{int}, n_1, n_2 分别为母节点和两个子节点的输入事例数.

在实际的训练过程中，一般将每个变量 $(x_1, x_2, \cdots, x_n)$ 的值域分为 n_{cuts} 个小区间，这 n_{cuts} 个区间的中心值作为 n_{cuts} 个阈值对增量 ΔI 进行计算，取其中的最大增量作为该变量的最大增量. 在所有 n 个变量 $(x_1, x_2, \cdots, x_n)$ 的最大增量中，数值最大的那个变量 x_j 作为本节点的判别变量，其最大增量对应的阈值 x_j^{th} 与 x_j 一起构成该节点的最优 (变量 + 阈值) 组合. 经验表明 n_{cuts} 取为 20 是个比较适当的选择，它是计算量和精细程度之间的一个比较适当的平衡，过大的 n_{cuts} 值并不能提升二叉树的信号/本底判别性能，反而不必要地增加了计算量.

显然训练的终结条件决定了二叉树的长度. 文献 [3] 提供了终止训练过程的几种方法. 经常实际使用的做法之一是设定一个最大的叶节点数，当训练过程已经形成的叶节点数等于大于该数值则训练停止. 另一种常用的方法是设定一个最小的事例数 N_{L}，当输入事例数小于 N_{L}，该节点的训练停止. 以上两种做法看起来缺乏理论依据，并且对于不同的问题需要根据经验确定适当的具体数值. 第三种做法是当一个节点的输入事例为同一类事例时，该节点的训练终止. 第四种做法是根据所有节点的增量值来决定训练是否终止，当节点增量 ΔI 满足

$$\Delta I \leqslant \beta, \qquad \beta > 0\text{常数} \tag{4.2.7}$$

则该节点的训练终止.

训练完成后，输入事例数中信号事例占优的叶节点被指定为二叉树的信号叶节点，本底事例占优的叶节点被指定为二叉树的本底叶节点. 这样一个二叉树就构建完成. 当一个待分类的事例样本集输入这样构建的二叉树后，归入信号叶节点的事例被判为“信号事例”，归入本底叶节点的事例被判为“本底事例”.

4.2.3 决策树的修剪

4.2.2 小节中讨论二叉树的构建时提到，利用一个训练样本集，从根节点开始，在每一个节点，通过某种优化步骤，寻找特征向量 $\boldsymbol{x}$ 的某一个变量及其阈值，使得在这一节点的判选中能最有效地区分信号和本底. 这一过程一直延续下去，直到满足某种终结条件为止，完成二叉树的构建. 例如可以进行到每个子节点只包含信号事例或本底事例为止. 这样构建的二叉树其节点数达到极大值. 初看起来，这种做法能够达到对信号和本底的错误率较低的判别，事实上，这种做法存在两个问题. 第一，起初决策树的错误率随节点数的增加而减小，但存在一个最佳节点数的决策树，它的错误率达到极小；当决策树的节点数大于该值时，错误率反而增加，所以决策树的节点数并非越多越好. 第二，过长的决策树训练得到的名义误判率往往低于误判率的真实值. 这种导致低估误判率的分叉过长的决策树训练 (构建) 称为过度训练.

L.Breiman 等认为[19]，利用某种终结条件构建决策树不是解决决策树最优化的正确途径. 寻找最优化决策树的正确方法是首先构建一个节点数达到极大的决策

树, 然后, 为了避免过度训练和计算量的有害增长, 需要进行修剪 (pruning). 所谓修剪, 就是对于节点数达到极大值的决策树自下而上地剪除对于有效地分辨信号/本底用处不大的节点. 显然, 需要确定一种准则来判断什么样的节点应当被剪除.

由 L.Breiman 等[19] 提出的最小复合费用修剪方案如下. 在二叉树的每个节点, 训练样本的误判率由式 (4.2.3) 定义:

$$R = 1 - \max(p, 1-p) \tag{4.2.8}$$

该节点的复合费用定义为

$$\rho = \frac{R - R_{\text{sub}}}{N_{\text{sub}} - 1}, \tag{4.2.9}$$

其中, R_{sub} 表示该节点以下的那部分二叉树的总误判率 (等于该二叉树所有叶节点的误判率之和); N_{sub} 表示该节点以下的那部分二叉树包含的叶节点数. 每个叶节点的信号纯度 p 等于到达该节点的信号事例数除以到达该节点的总事例数. 叶节点的误判率仍用式 (4.2.8) 计算. 一棵二叉树中, 假定复合费用 ρ 最小的节点称为节点 t, 当它的复合费用小于给定的修剪量 (prune strength)ρ_{PS}, 即

$$\rho_t < \rho_{\text{PS}}, \tag{4.2.10}$$

则节点 t 以下的部分二叉树被剪除, 而节点 t 变成一个 "新的" 叶节点. 这种修剪不断地进行, 直到不再出现这样的叶节点为止, 整棵二叉树的修剪得以完成.

修剪量的大小可以用下述步骤来确定: 将训练样本集分为两个子集, 子集 1 专用于构建二叉树, 子集 2 专用于构建完成的二叉树的性能测试. 对于一个给定的 ρ_{PS} 值, 用子集 1 构建二叉树 $T_1(\rho_{\text{PS}})$, 子集 2 进行其性能测试, 这样就得到二叉树性能与 ρ_{PS} 值的函数关系. 例如, 将子集 2 的 N_2 个样本输入 $T_1(\rho_{\text{PS}})$, 由于子集 2 的 N_2 个样本的分类是事先指定的, 因此 $T_1(\rho_{\text{PS}})$ 对这 N_2 个样本的分类错误的情况也能知道, 假定分类错误的样本数记为 $N_{\text{e}}(\rho_{\text{PS}})$ 个, 则 $T_1(\rho_{\text{PS}})$ 的错误率为

$$\varepsilon(T_1) = \frac{N_{\text{e}}(\rho_{\text{PS}})}{N_2}, \tag{4.2.11}$$

这样, 对于不同的 ρ_{PS} 值, 就得到 $\varepsilon(T_1)$ 与 ρ_{PS} 的函数关系, 可取 $\varepsilon(T_1)$ 最小的二叉树对应的 ρ_{PS} 作为最优的修剪量. 为了保证这种函数关系有较好的平稳性和统计稳定性, 两个子集的样本数应该足够大. 如果不作这样的优化, 一般可取 5 作为 ρ_{PS} 值.

4.3　决策树林法

决策树林法 (boosted decision trees)[22] 是决策树方法的扩展, 是为了克服决策树法对于训练样本集的统计涨落具有不稳定性的缺点而发展起来的, 它已证明是一

种有效而可靠和性能优良的分类器, 但到目前为止在高能物理实验中的实际应用还不是很多.

决策树法对于训练样本集的统计涨落具有不稳定性. 例如特征变量 x_1 和 x_2 有相近的信号/本底判别能力. 如果不存在统计涨落 (无限大样本集), 假设 x_1 比 x_2 有较强的判别能力, 所以会首先选择 x_1 来构建决策树; 但由于训练样本集的统计涨落 (有限样本集), 可能会首先选择 x_2 来构建决策树. 这两种不同结构的决策树对于同一个待分类事例可能给出不同的判别结果.

4.3.1 决策树林的构建

决策树法的统计不稳定性这一缺点在决策树林法中可得到克服. 它的基本思想是, 对于一个训练样本集, 构建第一棵决策树后, 对该样本集中的每个事例按某种规则赋予新的权值 (构建第一棵决策树时, 该样本集中的每个事例权值相等), 然后用具有新权值的样本集构建第二棵决策树, ……, 依次构建 K 棵决策树组成的树林. 当对任一新的事例作分类时, K 棵决策树组成的树林对于该事例的类别有 K 个判别结果, 以多数的结果作为整个决策树林对该事例类别的最终判决. 决策树林对同一个待分类事例给出不同结果的可能性较之单个决策树大大减小, 即在相当大的程度上克服了决策树法的统计不稳定性.

由以上讨论可见, 构建决策树林的关键点在于构建 K 棵决策树时怎样改变权值. 最常用的方法是 Y.Freund 和 R.E.Schapire 提出的自适应方法 (adaptive boost)[23], 其基本思想是在一棵决策树训练过程中被误判的所有事例在下一棵树的构建中赋以较高的权值, 判别正确的事例则保持权值不变.

假定训练样本集包含 N 个事例, 定义构建第 k 棵决策树时, 样本集中第 i 个事例 $\boldsymbol{x}_i$ 的权值为 $w_i(k), i=1,2,\cdots,N; k=1,2,\cdots,K$. 构建第一棵决策树时, 样本集中所有事例的权值均为 1, 即

$$w_i(1)=1, \qquad i=1,2,\cdots,N. \tag{4.3.1}$$

我们用 4.2 节讨论的二元决策树来训练样本集的 N 个事例. 用 $t_k(\boldsymbol{x}_i)$ 表示第 k 棵决策树 T_k 对于第 i 个事例 $\boldsymbol{x}_i$ 的判别结果的正确性, 即

$$\begin{cases} t_k(\boldsymbol{x}_i)=0, & \text{当对事例 } \boldsymbol{x}_i \text{ 判别正确;} \\ t_k(\boldsymbol{x}_i)=1, & \text{当对事例 } \boldsymbol{x}_i \text{ 判别错误.} \end{cases} \tag{4.3.2}$$

定义决策树 T_k 对于训练样本集 N 个事例的误判率为

$$\varepsilon_k=\sum_{i=1}^{N} w_i(k) t_k(\boldsymbol{x}_i) \Big/ \sum_{i=1}^{N} w_i(k), \qquad k=1,2,\cdots,K. \tag{4.3.3}$$

则构建决策树 T_{k+1} 时, 样本集中第 i 个事例的权值修改为

$$w_i^*(k+1) = w_i(k) \cdot \alpha_i(k). \tag{4.3.4}$$

其中

$$\begin{cases} \alpha_i(k) = \dfrac{1-\varepsilon_k}{\varepsilon_k} \equiv \alpha(k), & \text{事例 } i \text{ 被 } T_k \text{ 错误判别;} \\ \alpha_i(k) = 1, & \text{事例 } i \text{ 被 } T_k \text{ 正确判别.} \end{cases} \tag{4.3.5}$$

定义归一化常数 N_{k+1}^* 为权值和:

$$N_{k+1}^* = \sum_{i=1}^{N} w_i^*(k+1) = \sum_{i=1}^{N} w_i(k) \cdot \alpha_i(k) \tag{4.3.6}$$

则构建第 T_{k+1} 时, 样本集中第 i 个事例的归一化权值为

$$w_i(k+1) = w_i^*(k+1) \cdot \frac{N}{N_{k+1}^*} = N \cdot \frac{w_i(k) \cdot \alpha_i(k)}{\sum\limits_{i=1}^{N} w_i(k) \cdot \alpha_i(k)}. \tag{4.3.7}$$

这样, 归一化权值之和就等于训练样本集的事例总数:

$$\sum_{i=1}^{N} w_i(k+1) = \sum_{i=1}^{N} N \cdot \frac{w_i(k) \cdot \alpha_i(k)}{\sum\limits_{i=1}^{N} w_i(k) \cdot \alpha_i(k)} = N. \tag{4.3.8}$$

于是利用归一化权值 $w_i(k+1)$ 来构建决策树 T_{k+1}. 该过程一直进行到所有 K 棵决策树构建完成为止, 一个完整的决策树林的训练 (构建) 过程便完成了. 典型的 K 值为 1000~2000[24].

4.3.2 决策树林对输入事例的分类

(1) 用 K 个决策树的二元决策确定决策树林的输出

当对一个待分类事例进行判别时, 令该事例的特征向量为 $\boldsymbol{x}$, 将该事例输入构建完成的决策树林, 其中 $T_k(k=1,2,\cdots,K)$ 对事例的判定结果用 $h_k(\boldsymbol{x})$ 表示, 若判定为 “信号”, 则 $h_k(\boldsymbol{x})=1$; 若判定为 “本底”, 则 $h_k(\boldsymbol{x})=-1$. 整个决策树林对输入事例的输出 $y(\boldsymbol{x})$ 可取为 K 棵决策树输出值的简单平均:

$$y(\boldsymbol{x}) = \sum_{k=1}^{K} h_k(\boldsymbol{x})/K \tag{4.3.9}$$

或取为 K 棵决策树输出值的加权平均:

$$y(\boldsymbol{x}) = \sum_{k=1}^{K} \ln(\alpha_k) \cdot h_k(\boldsymbol{x}). \tag{4.3.10}$$

决策树林对输入事例的分类为

$$\begin{cases} \text{信号事例}, & \text{当 } y(\boldsymbol{x}) \geqslant y_{\text{th}}; \\ \text{本底事例}, & \text{当 } y(\boldsymbol{x}) < y_{\text{th}}. \end{cases} \tag{4.3.11}$$

式中, y_{th} 是事先给定的常数.

(2) 用训练纯度确定决策树林的输出

当用 4.2.2 小节的方法构建二叉决策树时, 对于每个叶节点, 其信号纯度 p 等于到达该节点的信号事例数除以到达该节点的总事例数. 当对一个待分类事例进行判别时, 如若该事例在决策树 $T_k(k=1,2,\cdots,K)$ 中最后落入的叶节点的训练纯度为 $p_k(\boldsymbol{x})$, 则整个决策树林对输入事例的输出 $y(\boldsymbol{x})$ 可取为 K 棵决策树输出值的加权平均:

$$y(\boldsymbol{x}) = \sum_{k=1}^{K} p_k(\boldsymbol{x}) h_k(\boldsymbol{x}). \tag{4.3.12}$$

因为叶节点的训练纯度对于过度训练是敏感的, 因此训练纯度 $p_k(\boldsymbol{x})$ 往往被过度估计. 因此使用这种方法必须加以小心. 迄今为止对于该方法的测试表明, 它的分类性能并不比方法 (1) 有明显改善.

4.3.3 重抽样法构建决策树林

在 4.3.1 小节中, 通过改变训练样本集中每个事例的权值来构造 K 棵决策树, 从而完成决策树林的构建. 其中权值的改变取决于上一棵决策树的误判率. 所谓的重抽样方法, 除了构建第一棵决策树时, 训练样本集中每个事例的权值都为 1 ; 在构建其余 $K-1$ 棵决策树时, 训练样本集中每个事例的权值是各自独立地、随机地确定的 (如用 [0~1] 区间的均匀随机数产生). 当然, 每棵决策树的权值之和需要归一化, 即归一化权值之和等于训练样本集的事例总数 N. 所以除了权值的修正方案不同, 利用重抽样方法构建决策树林的步骤与 4.3.1 小节的一般方法是相同的. 利用这种方法构建的决策树林对于训练样本事例的统计涨落具有比较好的稳定性.

第五章 人工神经网络

5.1 概 述

人的大脑是自然界造就的最高级产物. 人类大脑的思维是人类智能的集中体现. 人的大脑是由大约 10^{11} 数量级的神经元和 $10^{14} \sim 10^{15}$ 数量级的突触组成的复杂系统. 人工神经网络是对人脑神经网络的结构、特性以及功能进行理论抽象、简化和模拟而构建的一种信息处理系统. 从系统的观点看, 人工神经网络是由大量神经元通过丰富和完善的连接而构成的自适应非线性动态系统. 自 1943 年以来, 人工神经网络在理论和实践两方面都取得了很大进展. 当前神经网络的应用已经渗透到多个领域, 如模式识别、智能控制、信号处理、优化计算、计算机视觉、生物医学工程等. 本章简略介绍人工神经网络中与实验数据多元分析相关的基本内容. 对于人工神经网络的更广泛和深入的了解可参考有关的文献和书籍[25~29]. 人工神经网络在粒子物理实验数据分析中的应用可参考文献 [30].

5.1.1 生物神经元和人工神经元

神经元 (即神经细胞) 是大脑处理信息的基本单元, 它的基本结构如图 5.1 所示. 神经元由 4 个部分组成: 细胞体、树突、轴突和突触. 细胞体是神经元新陈代谢的中心, 是接受与处理信息的部件. 树突是神经元的输入通道, 接受来自其他神经元的信息. 轴突是神经元的输出通道, 用于输出神经元的脉冲信号, 轴突远端的分支可与多个神经元连接. 一个神经元的神经末梢与另一神经元树突或细胞体的接触处称为突触, 它是神经元之间传递信息的输入输出接口. 一般, 神经元的脉冲信号经树突的突触传到下一个神经元导致其兴奋, 而经细胞体的突触传到下一个神经元导致其抑制.

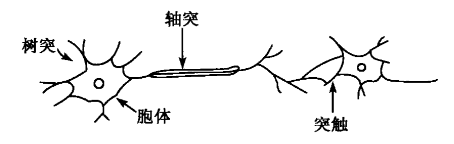


图 5.1 生物神经元结构示意图

神经元的基本工作机制是这样的: 一个神经元有两种状态 —— 兴奋和抑制. 平时处于抑制状态的神经元, 其树突和细胞体接受其他神经元传来的输出脉冲信

息, 多个输入在神经元中以代数和的方式叠加. 如果兴奋总量超过某个阈值, 该神经元就被激发进入兴奋状态, 并向外发出输出脉冲, 通过轴突传递给其他神经元. 神经元被激发之后有一个不应期, 在此期间不能再被激发, 相当于阈值电位突然升高, 然后阈值逐渐下降, 恢复其被激发的活性. 当然, 以上关于神经元工作机制的描述是极度简化的. 归纳起来, 生物神经元具有以下几个特性:

(1) 神经元是一个多输入、单输出的非线性信息处理单元.

(2) 神经元的输出响应是所有输入的综合累加作用的结果, 输入或输出分为兴奋型 (正值) 和抑制型 (负值) 两种.

(3) 神经元具有可塑性, 表现为其输出强度是可调节的.

人工神经元是一个数学模型, 模拟生物神经元的信息传递和处理功能. 人工神经元模型应当能够体现生物神经元的上述特征. 人工神经元模型种类繁多, 这里只介绍常用的最简单的一种模型, 即 1943 年由美国心理学家 McCulloch 和数学家 Pitts 提出的形式神经元的数学模型, 简称为 MP 模型[31]. 它的工作原理如图 5.2 所示.

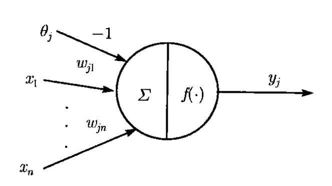


图 5.2 人工神经元 MP 模型

图 5.2 中 n 个输入 $x_1, \cdots, x_n$ 相当于其他 n 个神经元对于神经元 j 的输入值, n 个权值 $w_{j1}, \cdots, w_{jn}$ 相当于突触的连接强度, Σ 表示该神经元对于 n 个输入信号的累加, f 表示神经元 j 对于 n 个输入信号的响应, 称为变换函数或激活函数. θ_j 是该神经元的阈值. 采用如下记号

$$net_j = \sum_{i=1}^{n} w_{ji}x_i - \theta_j = \sum_{i=0}^{n} w_{ji}x_i \qquad (x_0 = \theta_j, w_{ji} = -1) \tag{5.1.1}$$

则神经元 j 的输出值 y_j 可表示为

$$y_j = f(net_j) \tag{5.1.2}$$

常见的变换函数有

(a) 线性函数

$$f(s) = s \tag{5.1.3}$$

(b) 符号函数

$$f(s) = \mathrm{sgn}(s) = \begin{cases} 1, & s \geqslant 0 \\ -1, & s < 0 \end{cases} \tag{5.1.4}$$

(c) 饱和函数

$$f(s)=\begin{cases}1, & s\geqslant \dfrac{1}{k}\\ ks, & -\dfrac{1}{k}\leqslant s<\dfrac{1}{k}\\ -1, & s<-\dfrac{1}{k}\end{cases} \tag{5.1.5}$$

(d) 双曲线正切函数

$$f(s)=\mathrm{th}(s)=\frac{1-\mathrm{e}^{-s}}{1+\mathrm{e}^{-s}}=\frac{2}{1+\mathrm{e}^{-s}}-1 \tag{5.1.6}$$

(e) 阶跃函数

$$f(s)=\begin{cases}1, & s\geqslant 0\\ 0, & s<0\end{cases} \tag{5.1.7}$$

(f) Sigmoid 函数

$$f(s)=\frac{1}{1+\mathrm{e}^{-s}} \tag{5.1.8}$$

以上这些变换函数的图形如图 5.3 所示.

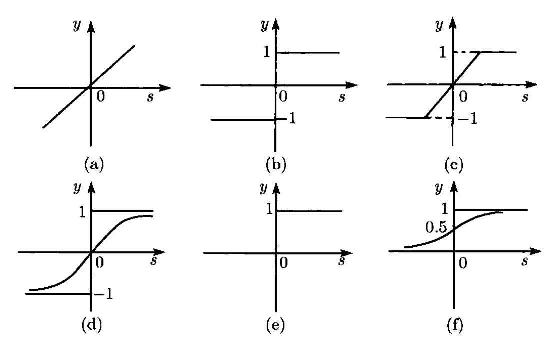


图 5.3　常用的变换函数

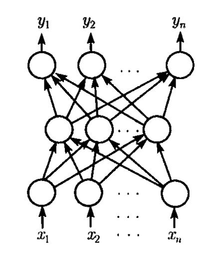
一些重要的神经网络算法要求变换函数 f 可微, 这时通常选用 S 型函数, 即 Sigmoid 函数和双曲线正切函数. 选择 S 型函数作为输出函数是由于它具有以下有用的特性: 非线性单调函数, 无限次可微, 权值很大时逼近阈值函数, 权值很小时逼近线性函数.

5.1.2　人工神经网络的构成和学习规则

大脑神经网络系统之所以具有思维认识等高级智能, 是由于它是由大量神经元相互连接而构成的一个复杂的神经网络系统. 人工神经网络也一样, 单个神经元的

功能极其有限, 只有许多神经元按一定的方式连接构成的神经网络才具有强大的功能. 与生物神经网络中神经元数量庞大、神经元结构有所差别、神经元的连接方式具有不同的形态相比, 人工神经网络是由数量远少于前者的、结构相同的神经元按一定规律构成的网络, 这种简化很大程度上是由于完全模拟的物理困难和计算的简便.

人工神经网络的连接形式其拓扑结构可以有很多种类, 但大致可以归纳为图 5.4 所示的两种形式: 阶层型和全互连型. 阶层型网络中的每一个神经元只能与相邻层的神经元发生相互作用, 而与本层的其他神经元不发生信息传递. 全互连型网络中的每一个神经元可与其他所有的神经元发生相互作用. 阶层型网络的层数和各层神经元的个数根据要求可以不同, 全互连型网络的神经元个数也可根据要求有所不同.



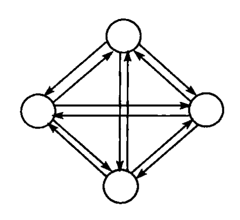


图 5.4　人工神经网络的连接形式

(a) 阶层型; (b) 全互连型

一个神经网络的拓扑结构确定之后, 为了使它具有某种智能特性, 还必须有相应的学习 (或训练) 规则与之配合.

对于大脑神经系统而言, 不同的功能区域均有各自的学习规则, 这些巧妙而完善的学习规则是大脑在进化过程中通过学习得到的. 对于人工神经网络而言, 其学习问题归根结底就是网络连接权的调整问题. 网络连接权的确定通常有两种方法. 一种是根据问题的具体要求直接计算, 后面要讨论的 Hopfield 网络作优化计算时就属于这种情况; 另一种是通过学习得到的, 大多数神经网络都使用这种方法. 其学习规则可由图 5.5 描述. 由式 (5.1.1)~(5.1.2) 可知, 网络对模式的判断取决于神经元的输出, 即取决于神经元的连接权, 当将一个给定初始连接权值的网络应用于模式识别时, 如果网络给出正确的

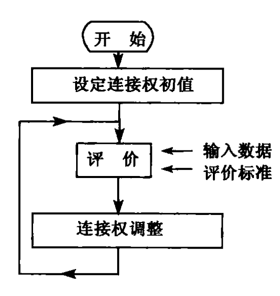


图 5.5　学习过程示意图

判断, 这样的行为应该得到增强 (提高权值); 反之如果网络给出错误的判断, 这样的行为应该减少 (减小权值). 这样的连接权的调整经过大量次数的重复之后, 网络对于模式的记忆就分布在网络所有神经元的连接权上. 当网络再对任意一个模式进行判别时就能进行正确的识别.

从网络学习过程的方式而言, 可以分为有教师 (或有监督) 学习和无教师 (或无监督) 学习. 在有教师学习方式中, 对于网络的学习结果, 即网络输出的正确性必须有一个评价标准, 网络根据实际输出与评价标准的比较, 决定连接权的调整量. 这个评价标准是人为地从外界提供给网络的, 相当于有一位知晓正确结果的教师示教给网络. 这种有教师学习方式的原理如图 5.6(a) 所示. 另一种重要的学习方式是无教师学习, 它是一种自组织学习, 即网络的学习过程完全是一种自我学习过程, 不存在外部教师的示教, 网络能够根据其特有的网络结构对属于同一类的模式进行自动分类, 可以认为, 这种网络的学习评价标准隐含于网络的内部. 无教师学习方式的原理如图 5.6(b) 所示.

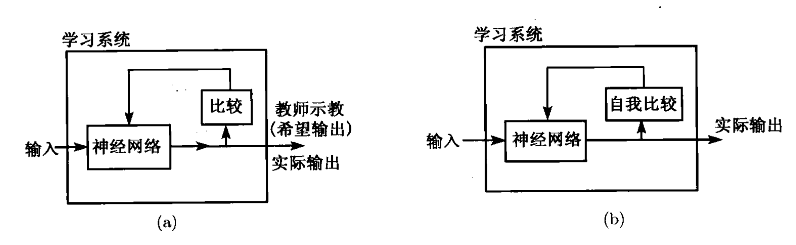


图 5.6

(a) 有教师学习; (b) 无教师学习

网络的学习规则是多种多样的, 但几乎所有神经网络的学习规则都可以看作 Hebb 规则的变形. 所谓 Hebb 规则, 是 Donall Hebb 根据生理学中条件反射机理于 1949 年提出的神经元连接强度变化的规则, Hebb 规则假定, 当两个神经元同时兴奋时, 它们之间的连接强度应该增加. 在人工神经网络中 Hebb 算法可描述为: 如果神经元 j 接受来自神经元 i 的输出, 则当这两个神经元同时兴奋时, 它们之间的连接权就应当增强, 用数学公式表示为

$$\Delta w_{ji} = w_{ji}(t+1) - w_{ji}(t) = \eta y_j x_i \tag{5.1.9}$$

式中, x_i 为神经元 i 的输出; y_j 为神经元 j 的输出; $w_{ji}(t)$ 为第 t+1 次调节前神经元 i 和 j 之间的连接权值; $w_{ji}(t+1)$ 为第 t+1 次调节后神经元 i 和 j 之间的连接权值; Δw_{ji} 为连接权的调整量; η 为学习 (或训练) 速率系数.

无论哪种形式的神经网络都有一个共同的特点: 网络的学习和工作运行取决

于各神经元连接权的动态演化过程. 某些拓扑结构相同的神经网络会具有不同的功能和特性, 是因为它们具有不同的学习和工作规则, 即不同的连接权的动态演化规则. 可见决定一个网络性质的主要因素有两点: 一是网络的拓扑结构, 另一个是网络的学习和工作规则, 两者结合起来构成一个网络的主要特征.

5.2 感 知 器

5.2.1 单输出单元感知器

美国学者 F.Rosenblatt 于 1957 年在 MP 模型和 Hebb 学习规则的基础上提出了具有自学习能力的感知器 (perceptron) 模型[32].

最简单的感知器的结构如图 5.7 所示. 它相当于一个具有 n 个输入节点的神经元, n 个输入 $x_1,\cdots,x_n$ 以相应的权值 $w_1,\cdots,w_n$ 输入计算单元, 通过一个符号函数式 (5.1.4) 作用后, 给出输出信息. 从数学上说, 即其输入加权和大于等于阈值时, 输出为 1 ; 否则为 -1. 用公式表示即输出值 y 为

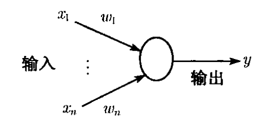


图 5.7 感知器结构示意图

$$y = \operatorname{sgn}\left(\sum_{i=1}^{n} w_i x_i - \theta\right). \tag{5.2.1}$$

单个神经元感知器与 MP 模型的不同之处在于神经元之间的耦合程度 (即连接权向量) 可变, 这样它就具有学习功能了. 网络学习的目的是对两类输入模式进行正确的分类, 即通过对模式样本的学习, 能够对输入模式进行 "0 ", "1" 分类.

假定我们已有了已知两种模式的 N 个训练样本, 网络按如下规则进行学习.

(1) 设置初值: 将权向量 $w_i(t=0), i=1,\cdots,n$ 和阈值 θ 赋予 $(-1,+1)$ 区间内的随机值. 这里 $w_i(t)$ 表示 t+1 次修正前第 $i(i=1,2,\cdots,n)$ 个输入节点与计算单元间的连接权.

(2) 输入一个样本 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$ 和它的期望输出 (教师示教)$\hat{y}$.

(3) 计算实际输出 $y=\operatorname{sgn}\left(\sum_{i=1}^{n} w_i x_i - \theta\right)$.

(4) 修正权向量 $\boldsymbol{w}$:

$$w_i(t+1) = w_i(t) + \Delta w_i(t), \quad i=0,1,\cdots,n. \tag{5.2.2}$$

和阈值

$$\theta(t+1) = \theta(t) + \Delta\theta(t) \equiv \theta(t) + \eta\delta(t), \quad i=0,1,\cdots,n. \tag{5.2.3}$$

其中

$$\begin{cases} \Delta w_i(t) = \eta x_i\,[\hat{y} - y(t)] \equiv \eta x_i \delta(t) \\ \Delta\theta(t) = \eta\,[\hat{y} - y(t)] \equiv \eta\delta(t) \end{cases}. \tag{5.2.4}$$

$\delta(t) = \hat{y} - y(t)$ 反映期望输出值与实际输出值之间的误差, 也称为学习信号.

(5) 回到第 (2) 步, 对于所有 N 个训练样本反复运用步骤 (3)~(4), 直到权向量 $\boldsymbol{w}$ 和阈值稳定不变为止, 学习过程结束.

学习率 η 通常取值为 $0 < \eta < 1$, 用于控制修正速度. η 太小, 权向量 $\boldsymbol{w}$ 收敛太慢；η 太大, 会导致权向量 $\boldsymbol{w}$ 不稳定. 所谓期望输出值 $\hat{y}$ 是对样本的一种人为分类. 比如对于这里的两类模式判别问题, 可以规定模式 “0 ”, “1” 的期望输出值 $\hat{y}$ 分别为 -1 和 $+1$. 于是有

$$\delta(t) = \hat{y} - y(t) = \begin{cases} 2, & \text{当}\hat{y} = 1,\ \ y(t) = -1 \\ 0, & \text{当}\hat{y} = y(t) \\ -2, & \text{当}\hat{y} = -1,\ \ y(t) = 1 \end{cases}. \tag{5.2.5}$$

上式说明, 当 $\hat{y} = y(t)$, 期望输出值与实际输出值相等, 连接权不需要调整. 当 $\hat{y} \neq y(t)$ 且误差 $\delta(t) > 0$, 说明权值 w_i 太小, 权值向增大方向调整. 当 $\hat{y} \neq y(t)$ 且误差 $\delta(t) < 0$, 说明权值 w_i 太大, 权值向减小方向调整. 这种按照期望输出与实际输出的误差 δ 来调节连接权强度的方法称为 δ 规则.

学习结束后, 网络将训练样本的模式以连接权 $\boldsymbol{w} = (w_1, w_2, \cdots, w_n)^{\mathrm{T}}$ 和阈值 θ 的形式 “记忆” 下来. 当给网络提供任意输入样本时, 网络按该样本的特征向量值 $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}$ 和记住的连接权 $\boldsymbol{w} = (w_1, w_2, \cdots, w_n)^{\mathrm{T}}$ 和阈值 θ 计算输出值 y, 并根据 y 为 $+1$ 或 -1 判断该样本属于记忆中的哪一种模式. 这一过程称为回想过程.

5.2.2　多输出单元感知器

多输出单元感知器由输入层和输出层组成, 输入层的神经元为输入节点, 输入节点的个数与模式特征变量维数 n 相等, 直接将 n 个变量值传输到下一层的各个神经元. 输出层的神经元为计算单元, 每个计算单元可接受前一层所有节点的输入, 但只有一个输出, 输出层神经元的个数等于待识别的模式的种类数 c. 同一层内的节点之间没有相互作用. 图 5.8 是多输出单元感知器的示意图.

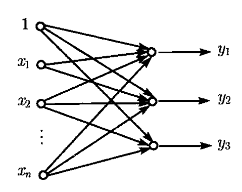


图 5.8　多输出单元感知器示意图

假定输入的增广特征向量为 $\boldsymbol{x} = (1, x_1, x_2, \cdots, x_n)^{\mathrm{T}}$. 输出层各单元的阈值为 $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_c)^{\mathrm{T}}$, 输入层与输出层各单元间的连接权为 $w_{ji}(j = 1, \cdots, c,\ i = 1, 2, \cdots, n)$ 是输入单元 i 和输出单元 j 之间的连

接权, $w_{j0}=-\theta_j(j=1,\cdots,c)$ 为输出层各单元的阈值. 因此增广权矩阵为

$$W_{c\times(n+1)}=\begin{pmatrix} w_{10} & w_{11} & w_{12} & \cdots & w_{1n} \\ w_{20} & w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ w_{c0} & w_{c1} & w_{c2} & \cdots & w_{cn} \end{pmatrix} \tag{5.2.6}$$

分类器学习的目标是通过调整权值使网络由给定的输入模式类得到给定的输出值. 用已知类别的样本集作为训练集, 当输入 j 类样本时, 使对应于该类的输出 $y_j=1$, 而其他计算单元的输出为 -1, 这是我们期望的输出值. 设期望的输出为

$$\hat{\boldsymbol{y}}=(\hat{y}_1,\hat{y}_2,\cdots,\hat{y}_c)^{\mathrm{T}}$$

根据输入特征向量计算得到的输出为

$$\boldsymbol{y}=(y_1,y_2,\cdots,y_c)^{\mathrm{T}}$$

其中

$$\left.\begin{aligned} y_j&=f(net_j) \\ net_j&=\sum_{i=0}^{n}w_{ji}x_i \end{aligned}\right\},\qquad j=1,2,\cdots,c \tag{5.2.7}$$

为了使计算得到的输出逼近期望的输出, 对权值和阈值作如下的调整

$$w_{ji}(t+1)=w_{ji}(t)+\Delta w_{ji}(t),\qquad i=0,1,\cdots,n,\quad j=1,\cdots,c. \tag{5.2.8}$$

其中

$$\Delta w_{ji}(t)=\eta x_i\left[\hat{y}_j-y_j(t)\right]\equiv\eta x_i\delta_j(t). \tag{5.2.9}$$

由此, 我们可得出双层感知器的学习规则. 假定我们已有了 N 个已知类别分别为 $m=1,2,\cdots,c$ 的训练样本, 网络按如下规则进行学习.

(1) 设置初值: 将连接权 $\boldsymbol{W}$ 各元素赋予 $(-1,+1)$ 区间内的随机值.

(2) 输入一个 m 类样本 $\boldsymbol{x}^m=(1,x_1^m,x_2^m,\cdots,x_n^m)^{\mathrm{T}}$ 和它的期望输出 (教师示教) $\hat{\boldsymbol{y}}^m=(\hat{y}_1^m,\hat{y}_2^m,\cdots,\hat{y}_c^m)^{\mathrm{T}}$, $\hat{y}_{j=m}^m=1$, $\hat{y}_{j\neq m}^m=-1$, $j=1,2,\cdots,c$.

(3) 计算输出层各单元输出

$$y_j^m=\mathrm{sgn}\left(\sum_{i=0}^{n}w_{ji}x_i^m\right),\qquad j=1,2,\cdots,c$$

(4) 修正连接权 $\boldsymbol{W}$ 各元素

$$w_{ji}(t+1)=w_{ji}(t)+\Delta w_{ji}(t),\qquad i=0,1,\cdots,n,\quad j=1,\cdots,c.$$

其中

$$\Delta w_{ji}(t)=\eta x_i^m\left[\hat{y}_j^m-y_j^m(t)\right]\equiv\eta x_i^m\delta_j^m(t).$$

(5) 回到第 (3) 步, 对于该样本 $\boldsymbol{x}^m$ 反复运用步骤 (3)~(4), 直到连接权 $\boldsymbol{W}$ 稳定不变为止.

(6) 回到第 (2) 步, 对于所有 N 个训练样本反复运用步骤 (2)~(5), 直到连接权 $\boldsymbol{W}$ 稳定不变为止, 学习过程结束.

Rosenblatt 从数学上给出了严格的证明 (参见文献 [32]), 对于线性可分的样本集, 感知器算法是收敛的, 就是说 $\boldsymbol{W}$ 一定存在, 并且学习过程在有限次迭代后得以完成. 对于非线性可分的样本集, 感知器算法会发生振荡, $\boldsymbol{W}$ 不收敛. 虽然感知器只能对线性可分的输入模式进行正确分类, 但它作为人工神经网络的初期模型, 特别是其自学习自组织的思想, 对于人工神经网络理论的研究和发展产生了深远的影响.

5.3　多层前向神经网络和误差逆传播算法

前向神经网络是一种层状结构的神经网络, 第一层为输入层, 最后一层为输出层, 中间各层为隐含层, 可以有多个隐含层. 输入层的神经元为输入节点, 其他各层为计算单元. 输入节点的个数与模式特征变量维数 n 相等, 直接将 n 个变量值传输到下一层的各个神经元. 每个计算单元可接受前一层所有节点的输入, 但只有一个输出, 该输出耦合到下一层的所有神经元. 同一层内的节点之间没有相互作用. 输出层神经元的个数等于待识别的模式的种类数 c. 对于 c=2 的两类模式识别问题, 输出层神经元的个数可以为 1, 其二值输出 (0,1) 或 (−1, +1) 表示两种不同模式的判别结果. 隐含层的节点数没有明确的规则可以遵循, 一般来说, 问题越复杂, 需要的单元数越多. 图 5.9 是一个四层前向神经网络的结构示意图.

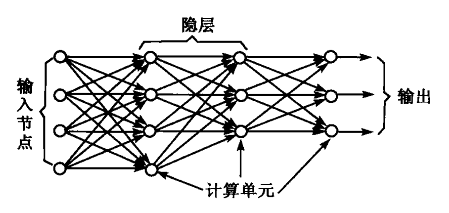


图 5.9　四层前向神经网络结构示意图

对于非线性可分的样本集, 感知器算法不收敛这一缺点, 利用包含隐含层的多层前向神经网络能够克服. 多层前馈网络的学习算法比较复杂, 其主要困难在于中间的隐含层不直接与网络的输出连接, 无法直接计算其误差. 为了解决这一问题, 提出了误差逆传播 (back-propogation, 简称 BP) 算法[33]. 其主要思想是从后向前(逆向) 传播输出层的误差, 以间接算出隐含层的输出误差. 算法分为两个阶段: 第一阶段 (正向过程), 输入信息从输入层经过隐含层到输出层逐层计算各单元的输出值 ; 第二阶段 (误差逆传播过程), 输出误差逐层从后向前算出隐含层各单元的输出误差, 并用此误差修正各层之间的权值. 利用误差逆传播算法的多层前馈网络常称为 BP 网络.

5.3.1 BP 网络学习算法

BP 网络中通常采用梯度法修正权值, 为此要求输出函数可微, 通常采用 Sigmoid 函数作为输出函数. 不失普遍性, 我们研究某一层第 j 个计算单元. 在下面的叙述中, 角标 i 代表其前一层的第 i 个单元, 角标 k 代表其后一层的第 k 个单元. O_j 代表本层单元 j 的输出, O_i 代表前一层单元 i 的输出. w_{ji} 代表本层单元 j 与前一层单元 i 间的权值.

当输入某个类别已知的样本的增广特征向量 $\boldsymbol{x} = (1, x_1, x_2, \cdots, x_n)^{\mathrm{T}}$ 时, 从前向后 (正向算法) 对各层单元计算其总输入 net_j 和输出 O_j

$$net_j = \sum_i w_{ji} O_i \tag{5.3.1}$$

$$O_j = f(net_j) \tag{5.3.2}$$

当考虑的是输出层的单元 j 时, 实际输出是 $y_j = O_j$. 假定此样本的期望输出用 $\hat{y}_j$ 表示, 网络对于此样本的输出误差 E 可表示为

$$E = \sum_j E_j = \sum_j \frac{1}{2}(\hat{y}_j - y_j)^2 \tag{5.3.3}$$

我们的目标是寻找一组各层 (包括隐含层和输出层) 的权矩阵, 使得误差目标函数 E 达到极小. 优化计算的方法很多, 其中常用的一种是一阶梯度法, 即最速下降法. 下面具体介绍这种方法.

由于单元 j 总输入 net_j 的变化导致样本误差 E 的变化可用梯度值表示

$$\delta_j \equiv \frac{\partial E}{\partial net_j} \tag{5.3.4}$$

网络学习的目的是求得适当的权值, 为此考虑单元 j 的权值 w_{ji} 的变化对样本误差 E 的影响, 可有

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}}$$

该式中

$$\frac{\partial net_j}{\partial w_{ji}} = \frac{\partial \sum_i w_{ji} O_i}{\partial w_{ji}} = O_i \tag{5.3.5}$$

是权值 w_{ji} 的变化对于 j 单元总输入 net_j 的变化的速率 (梯度). 因此可得

$$\frac{\partial E}{\partial w_{ji}} = \delta_j O_i \tag{5.3.6}$$

权值 w_{ji} 的修正应该向样本误差 E 减小的方向进行, 即向负梯度 $D \equiv -\frac{\partial E}{\partial w_{ji}} = -\delta_j O_i$ 方向进行, 因此权值的修正为

$$\Delta w_{ji} = -\eta \delta_j O_i \equiv \eta D \tag{5.3.7}$$

其中, η ($\eta > 0$) 是权值的修正系数.

$$w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji}(t) \tag{5.3.8}$$

如果节点 j 是输出单元, 则

$$\begin{aligned} O_j &= y_j \\ \delta_j &= \frac{\partial E}{\partial net_j} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial net_j} = -(\hat{y}_j - y_j) f'(net_j) \end{aligned} \tag{5.3.9}$$

如果节点 j 不是输出单元, 则节点 j 的输出 O_j 对后层的全部节点都有影响, 因此

$$\begin{aligned} \delta_j &= \frac{\partial E}{\partial net_j} = \sum_k \frac{\partial E}{\partial net_k} \cdot \frac{\partial net_k}{\partial O_j} \cdot \frac{\partial O_j}{\partial net_j} \\ &= \sum_k \delta_k w_{kj} f'(net_j) \end{aligned} \tag{5.3.10}$$

其中, 计算 $\frac{\partial O_j}{\partial net_j}$ 用到式 (5.3.2); 计算 $\frac{\partial net_k}{\partial O_j}$ 用到式 (5.3.1); δ_k 则由输出层的 δ 值自后向前逐层反推得到.

对于 Sigmoid 函数,

$$\begin{aligned} y &= f(s) = \frac{1}{1+\mathrm{e}^{-s}} \\ f'(s) &= \frac{\mathrm{e}^{-s}}{(1+\mathrm{e}^{-s})^2} = y(1-y) \end{aligned} \tag{5.3.11}$$

或者利用双曲线函数作为变换函数, 则有

$$\begin{aligned} y &= f(s) = \mathrm{th}s \\ f'(s) &= 1 - \mathrm{th}^2 s = 1 - y^2 \end{aligned} \tag{5.3.12}$$

综上所述, 我们得到反向传播算法的学习规则如下. 假定我们已有了已知 c 种模式的 N 个训练样本, 网络按如下规则进行学习.

(1) 设置初值: 将连接权 $\boldsymbol{W}$ 各元素赋予 $(-1,+1)$ 区间内的随机值, 设定修正系数 η $(\eta>0)$, 设定收敛值 ε.

(2) 输入一个样本 $\boldsymbol{x}^m=(1,x_1^m,x_2^m,\cdots,x_n^m)^{\mathrm{T}}$ 和它的期望输出 (教师示教) $\hat{\boldsymbol{y}}^m=(\hat{y}_1^m,\hat{y}_2^m,\cdots,\hat{y}_c^m)^{\mathrm{T}}$. 并作如下计算

(a) 从前向后逐层计算各单元输出 O_j

$$net_j=\sum_i w_{ji}O_i$$
$$O_j=1/(1-\mathrm{e}^{-net_j})$$

(b) 计算输出 $\boldsymbol{y}^m$ 与期望输出 $\hat{\boldsymbol{y}}^m$ 的欧氏距离 d, 若 d 小于收敛值 ε, 即认为连接权 $\boldsymbol{W}$ 稳定, 执行第 3 步 ; 否则, 执行下一步.

(c) 从后向前逐层计算 δ_j

输出层 $$\delta_j=(\hat{y}_j^m-O_j)O_j(1-O_j)$$

隐含层 $$\delta_j=O_j(1-O_j)\sum_k w_{kj}\delta_k$$

(d) 计算并保存各权值修正量

$$\Delta w_{ji}(t)=-\eta\delta_j O_i$$

(e) 修正连接权 $\boldsymbol{W}$ 各元素

$$w_{ji}(t+1)=w_{ji}(t)+\Delta w_{ji}(t)$$

(f) 回到 (a).

(3) 回到第 (2) 步, 对于所有 N 个训练样本反复运用步骤 (2), 直到对所有样本欧氏距离 d 都小于收敛值 ε, 这时各层连接权 $\boldsymbol{W}$ 即为问题的解, 学习过程结束.

上述学习方法是对于每个训练样本逐个地进行权值修正的, 这种方法也称为标准的误差逆传播算法. 它的计算流程图如图 5.10 所示.

二层前馈网络的收敛性不受初始值影响. 三层以上的前馈网络使用误差逆传播算法时, 收敛性受初始值影响. 通常用较小的随机数 (例如 ±0.3 区间) 作为权值的初值. 当计算不收敛时, 可以改变初始值再试验.

BP 算法实质上是把一组样本的判别问题转化为一个非线性优化问题, 并通过梯度法利用迭代运算求解权值的一种学习方法. 已经证明, 利用 Sigmoid 函数作为变换函数的三层 BP 网络可以以任意精度逼近任意连续函数. 也就是说, 三层 BP 网络原则上可以解决任意非线性的分类问题. 这是 BP 网络的一个显著的优点, 也使得它在分类器中得到广泛的应用.

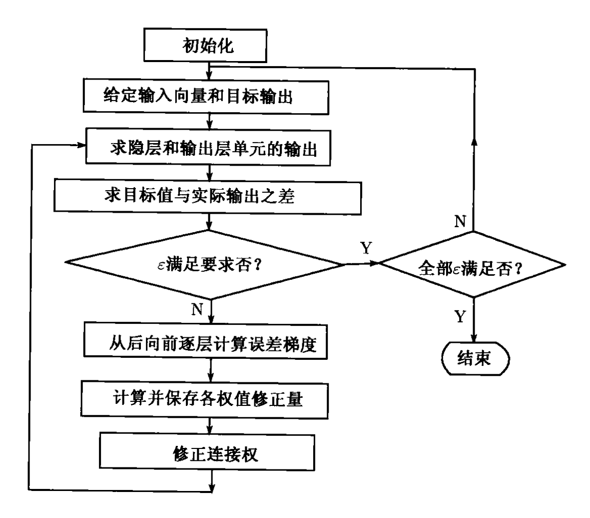


图 5.10　误差逆传播算法流程图

但是 BP 网络也存在以下一些缺点:

(1) 由于采用梯度算法, 容易陷入局部极小而得不到全局极小点. 能否收敛到全局极小往往取决于初始值.

(2) 决定收敛速度的权值修正系数 (学习率)η 的确定依赖于尝试和经验.

(3) 目标函数 E 是全体连接权的函数, 要寻优的参数 (各层的连接权矩阵的所有元素) 很多, 导致收敛速度慢.

(4) 对于三层 BP 网络, 其输入、输出层的节点数由问题本身决定, 输入层节点数等于特征向量维数, 输出层节点数等于模式类数. 但隐含层的节点数的确定缺乏理论指导和有效的方法. 对于三层以上的 BP 网络, 隐含层的数目和节点数的确定存在同样的困难.

5.3.2　BP 网络学习算法的改进

为了克服 BP 网络收敛速度慢的缺点, 对于 BP 网络学习算法的改进作了广泛的研究, 提出了许多改进方案. 下面介绍比较典型而且简便的几种.

1. *全局误差极小化方法*

标准的误差逆传播算法是使单个样本的误差函数 E 达到极小的一阶梯度法作优化计算的, 这种方法偏离了全局误差意义上的梯度下降. 因为我们要解决的是 c 个类别样本的分类问题, 所以对于权值进行调节时必须考虑 c 个类别样本的全局误

差.

假定分类器要区分 c 种类别, 当输入类别 $m(m=1,2,\cdots,c)$ 的一个训练样本, 可计算其误差函数 $E^{(m)}$, 全局误差函数 E 定义为

$$E=\sum_{m=1}^{c}E^{(m)}. \tag{5.3.13}$$

全局误差逆传播算法依靠对全局误差函数 E 作极小化计算推导, 因此需要将一组 c 个模式的训练样本输入网络后, 再调节权值. 这时 (5.3.7) 式的权值修正需修改为

$$\Delta w_{ji}=\eta D=-\eta\sum_{m=1}^{c}\delta_j^m O_i^m \tag{5.3.14}$$

由此可得全局误差逆传播算法的学习规则如下. 假定我们已有了 N 组已知类别分别为 $m=1,2,\cdots,c$ 的 c 个训练样本, 网络按如下规则进行学习.

(1) 设置初值: 将连接权 $\boldsymbol{W}$ 各元素赋予 $(-1,+1)$ 区间内的随机值, 设定修正系数 η $(\eta>0)$, 设定收敛值 ε.

(2) 依次输入一组 m $(m$=1,2,$\cdots,c)$ 类样本 $\boldsymbol{x}^m=(1,x_1^m,x_2^m,\cdots,x_n^m)^{\mathrm{T}}$ 和它的期望输出 (教师示教) $\hat{\boldsymbol{y}}^m=(\hat{y}_1^m,\hat{y}_2^m,\cdots,\hat{y}_c^m)^{\mathrm{T}}$, 并作如下计算

(a) 对每一个输入样本作如下计算

从前向后逐层计算各单元输出 O_j^m

$$net_j^m=\sum_i w_{ji}O_i^m$$

$$O_j^m=1/\left(1-\mathrm{e}^{-net_j^m}\right)$$

(b) 计算每个样本的输出 $\boldsymbol{y}^m$ 与期望输出 $\hat{\boldsymbol{y}}^m$ 的欧氏距离 d^m, 若该组 c 个样本的 $d^m<\varepsilon(m=1,2,\cdots,c)$, 即认为连接权 $\boldsymbol{W}$ 稳定, 执行第 (3) 步 ; 否则, 执行下一步.

(c) 从后向前逐层计算 δ_j^m

输出层 $\qquad \delta_j^m=(\hat{y}_j^m-O_j^m)O_j^m(1-O_j^m)$

隐含层 $\qquad \delta_j^m=O_j^m(1-O_j^m)\sum_k w_{kj}\delta_k^m$

(d) 对该组 c 个类型的样本作如下计算

计算并保存各权值修正量

$$\Delta w_{ji}(t)=-\eta\sum_{m=1}^{c}\delta_j^m O_i^m$$

修正连接权 $\boldsymbol{W}$ 各元素

$$w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji}(t).$$

回到步骤 (a).

(3) 回到第 (2) 步, 对于所有 N 组训练样本反复运用步骤 (2), 直到对所有样本欧氏距离 d 都小于收敛值 ε, 这时各层连接权 $\boldsymbol{W}$ 即为问题的解, 学习过程结束.

在网络的一次学习中, 全局误差逆传播算法中权值调整一次, 相当于标准的误差逆传播算法中权值调整 c 次, 因而权值的调整次数明显减少, 对于多类分类器学习时间大大缩短, 对于训练样本集不太大的情况, 收敛速度比较快. 但是这种算法将各种模式的误差平均化, 在有些情况下会引起网络的振荡.

2. 引入惯性修正项

所谓惯性修正项, 就是每一次对权值进行修正时, 按一定比例加上前一次的权值修正量, 即将式 (5.3.7) 修改为

$$\Delta w_{ji}(t) = \eta D(t) + \alpha \Delta w_{ji}(t-1) \tag{5.3.15}$$

其中, $\eta D(t) = -\eta \delta_j O_i$ 是本次修正量; $\Delta w_{ji}(t-1)$ 是上次修正量; α $(1 > \alpha > 0)$ 是惯性项的比例修正系数. 惯性项的引入实际上是考虑前一次权值修正时的梯度方向. 当上一次修正过量时, 惯性项与本次算得的修正量反号, 使得 $\Delta w_{ji}(t)$ 减小以减小振荡；当上一次修正欠量时, 惯性项与本次算得的修正量同号, 使得 $\Delta w_{ji}(t)$ 增大以加速收敛. 通常情况下, α 可取 0.9 或附近的值.

3. 变步长法

一阶梯度法寻优收敛较慢的一个重要原因是学习率 η 不好选择, η 太小, 收敛太慢；η 太大则可能过修正, 导致振荡甚至发散. 变步长法是针对该问题提出的改进方案. 权值的修正由下式表示

$$\begin{aligned} &\Delta w_{ji}(t) = \eta(t) D(t) \\ &\eta(t) = 2^{\lambda} \eta(t-1) \\ &\lambda = \operatorname{sgn}\left[D(t) D(t-1)\right] \end{aligned} \tag{5.3.16}$$

这样, 当连续两次修正中其梯度方向相同时 ($D(t)$ 与 $D(t-1)$ 同号), 表明修正量不足, 可使步长 η 加倍, 以加速收敛；当连续两次修正中其梯度方向相反时, 表明修正量过度, 可使步长 η 减半, 以避免振荡. 当需要引入惯性项时, 只需将上式中的 $\Delta w_{ji}(t)$ 用式 (5.3.15) 计算并将 η 用 $\eta(t)$ 计算即可. 当使用该算法时, 由于步长在迭代过程中自适应进行调整, 因此对于不同的连接权系数实际上采用了不同的学习率, 也就是说误差目标函数 E 在超曲面上在不同的方向按照各自比较合理的步长向极小点逼近.

5.4 Hopfield 神经网络

Hopfield 神经网络是美国物理学家 J.J. Hopfield[34] 于 1982 年首先提出的, 它主要用于模拟生物神经网络的记忆机理. 与前述的前馈网络不同, Hopfield 网络是一种全连接型网络. 网络的基本单元是与前馈网络类似的神经元, 它的结构是单层的, 各单元地位平等, 每个神经元与所有其他神经元连接. 对于每一个神经元而言, 自己的输出信号通过其他神经元又反馈到自己, 所以 Hopfield 神经网络是一种反馈型网络.

Hopfield 神经网络分为离散型 (DHNN) 和连续型 (CHNN) 两种.

Hopfield 神经网络状态的演变过程是一个非线性动力学动态过程, 可以用一组非线性差分方程 (对于 DHNN) 或微分方程 (对于 CHNN) 来描述. 系统的稳定性可用 "能量函数"(即李雅普诺夫或哈密顿函数) 进行分析. 在满足一定条件下, 能量函数在网络运行过程中不断减小, 最后趋于稳定态, 称为吸引子. 对于一个非线性动力学系统, 系统状态从某一初态出发, 经过演变后, 既可能到达稳定态, 也可能到达有界振荡态 (极限环) 、混沌态或发散. 但对于变换函数为有界函数的人工神经网络, 不会产生发散现象. Hopfield 神经网络在某些情况下还有随机性和不可预测性. 人们可以从不同的方面利用这些复杂的性质来完成各种计算功能.

5.4.1 离散 Hopfield 网络

1. *网络结构和工作方式*

离散 Hopfield 网络 DHNN 是一种单层的输入、输出均为二值的反馈网络, 主要用于联想记忆. DHNN 的结构如图 5.11 所示. 对于待分类的模式向量有 n 个分量的情形, 需用 n 个节点的离散 Hopfield 网络. 网络状态用向量 $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}$ 表示, 各分量是 n 个神经元的输出, 且 x_i 仅取 $+1$ 和 -1 两个值. $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_n)^{\mathrm{T}}$ 为 n 个神经元的阈值向量. $\boldsymbol{W} = [\boldsymbol{W}]_{n\times n}$ 为网络的连接权矩阵, 其元素 w_{ij} 表示神经元 i 和 j 之间的连接权. 权矩阵为对称矩阵, 即有 $w_{ij} = w_{ji}$. 若对角元素为 0, 即 $w_{ii} = 0$, 则称网络为无自反馈的. 以下讨论的均是无自反馈的离散 Hopfield 网络.

描写 DHNN 状态变化的方程如下

$$\begin{cases} u_i(t+1) = \displaystyle\sum_{j=1}^{n} w_{ij} x_j(t) - \theta_i \\ x_i(t+1) = \operatorname{sgn}\left[u_i(t+1)\right] \end{cases} \tag{5.4.1}$$

其中, $x_i(t)$ 为为任意时刻 t (t 为正整数) 神经元 i 的状态. DHNN 有两种工作方式.

(1) 串行 (异步) 方式. 在任一时刻, 只有某一个神经元 i(按固定顺序或随机选择) 按照式 (5.4.1) 改变状态, 其余神经元状态不变, 即

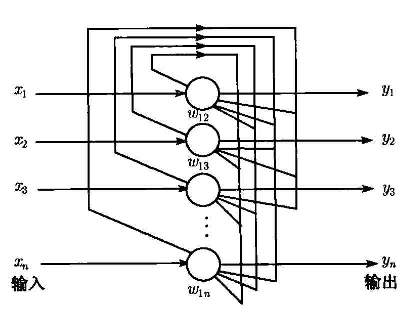


图 5.11　DHNN 结构

$$\begin{cases} x_i(t+1) = \operatorname{sgn}\left[\sum_{j=1}^{n} w_{ij}x_j(t) - \theta_i\right] \\ x_j(t+1) = x_j(t), \quad j \neq i \end{cases} \tag{5.4.2}$$

(2) 并行 (同步) 方式. 在任一时刻, 有部分神经元按照式 (5.4.1) 改变状态, 其余神经元状态不变. 其中最重要的一种特殊情况为所有神经元同时按照式 (5.4.1) 改变状态, 称为全并行方式, 这时有

$$x_i(t+1) = \operatorname{sgn}\left[\sum_{j=1}^{n} w_{ij}x_j(t) - \theta_i\right], \qquad i = 1, 2, \cdots, n \tag{5.4.3}$$

若网络从某一初态 $\boldsymbol{x}(0)$ 开始, 经过有限时间 t 后, 它的状态不再发生变化, 就达到了稳定态, 也称为吸引子, 用公式表示即为

$$x_i(t+1) = x_i(t) = \operatorname{sgn}\left[\sum_{j=1}^{n} w_{ij}x_j(t) - \theta_i\right], \qquad i = 1, 2, \cdots, n \tag{5.4.4}$$

若用向量记号, 也可写成

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{W}\boldsymbol{x} - \boldsymbol{\theta}) \tag{5.4.5}$$

式中, f 是 sgn 函数.

2. 网络稳定性和吸引子

从上述工作过程可以看出, DHNN 实质上是一个离散的非线性动力学系统. 如果系统是稳定的, 则它可从任一初态收敛到一个稳定态; 若系统是不稳定的, 由于网络节点输出只有 1 和 −1 两个值, 因此系统不可能出现无限发散只可能出现幅度为 2 的自持振荡, 或称为极限环.

为了研究网络的稳定性, 定义 DHNN 网络的能量函数 (或势函数)

$$E=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}w_{ij}x_ix_j+\sum_{i=1}^{n}\theta_ix_i \tag{5.4.6}$$

写成矩阵形式为

$$E=-\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{x}+\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\theta} \tag{5.4.7}$$

由于 x_i, x_j 只能为 ±1, w_{ij}, θ_i 有界, 因此能量函数 E 是有界的. 若从某一初始状态开始, 网络每次状态变化都能满足

$$\Delta E=E(t+1)-E(t)\leqslant 0 \tag{5.4.8}$$

即能量函数单调下降, 则网络状态最后趋于一个稳定点.

DHNN 网络工作于串行方式时式 (5.4.8) 成立. 证明如下: 当 DHNN 网络由时刻 t 到时刻 t+1, 只有一个神经元 i 的状态发生变化, 这时有

$$\begin{aligned}\Delta E=E(t+1)-E(t)=&-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}w_{ij}x_i(t+1)x_j(t+1)+\sum_{i=1}^{n}\theta_ix_i(t+1)\\&+\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}w_{ij}x_i(t)x_j(t)-\sum_{i=1}^{n}\theta_ix_i(t)\end{aligned}$$

考虑到 $w_{ij}=w_{ji}$,$w_{ii}=0$, 以及 $x_j(t+1)=x_j(t)$,$j\neq i$, 容易得到

$$\Delta E=E(t+1)-E(t)=-\left[x_i(t+1)-x_i(t)\right]\cdot\left[\sum_{j=1,j\neq i}^{n}w_{ij}x_j(t)-\theta_i\right] \tag{5.4.9}$$

由于 x_i, x_j 只能为 ±1, 故只需考虑以下 3 种情况:

(1) $x_i(t+1)=x_i(t)$, 由式 (5.4.9) 立即知 $\Delta E=0$.

(2) $x_i(t+1)-x_i(t)=2$, 即 $x_i(t+1)=1,x_i(t)=-1$, 由式 (5.4.3) 知 $\sum_{j=1}^{n}w_{ij}x_j(t)-\theta_i>0$, 注意到 $w_{ii}=0$, 故有 $\sum_{j=1,j\neq i}^{n}w_{ij}x_j(t)-\theta_i>0$, 即由式 (5.4.9) 知 $\Delta E<0$.

(3) $x_i(t+1)-x_i(t)=-2$, 即 $x_i(t+1)=-1,x_i(t)=1$, 由式 (5.4.3) 知

$\sum_{j=1}^{n} w_{ij}x_j(t) - \theta_i < 0$, 注意到 $w_{ii} = 0$, 故有 $\sum_{j=1, j\neq i}^{n} w_{ij}x_j(t) - \theta_i < 0$, 即由式 (5.4.9) 知 $\Delta E < 0$.

可见, 网络如果发生变化, 其势函数只可能减小. 注意到 n 个节点的 DHNN 网络只有有限个 (2^n) 状态, 最终一定会到达势函数的某一个极小点 (平衡态), 与该点相邻 (只某一个 x_i 不同) 的点的势函数值一定大于该点, 因此该平衡态是孤立的. 由于系统是非线性的, 可以有多个孤立平衡态. 从系统的状态空间的任何一点出发, 都会到达某个极小点, 好像被平衡态所吸引, 所以孤立平衡态又称为孤立吸引子. 到达某个吸引子的所有出发点的集合称为该吸引子的吸引域.

吸引子有如下性质: 如果状态向量 $\boldsymbol{x}$ 是网络的一个吸引子, 且阈值向量 $\boldsymbol{\theta} = 0, \sum_{j=1}^{n} w_{ij}x_j \neq 0$, 则 $-\boldsymbol{x}$ 也一定是网络的吸引子. 证明如下: 由于 $\boldsymbol{x}$ 是吸引子, 且 $\boldsymbol{\theta} = 0$, 由式 (5.4.5) 知 $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{W}\boldsymbol{x})$, 从而有 $\boldsymbol{f}[\boldsymbol{W}(-\boldsymbol{x})] = \boldsymbol{f}[-\boldsymbol{W}\boldsymbol{x}] = -\boldsymbol{f}[\boldsymbol{W}\boldsymbol{x}] = -\boldsymbol{x}$, 证毕.

对于全并行方式的 DHNN 网络, 可以证明, 若连接权矩阵为非负定对称矩阵, 则对任意初态, 网络收敛于一个孤立吸引子; 若连接权矩阵为负定对称矩阵, 则网络周期振荡, 极限环为 2.

上面讨论的是单元状态取值为$\{+1, -1\}$的情况. 对于单元状态取值$\{1, 0\}$的 DHNN 网络, 上述结论仍然成立.

由于异步方式比同步方式有更好的稳定性, 实际使用中多采用异步方式. 异步方式的缺点是失去了神经网络并行处理的优点.

3. 网络的联想记忆

从上述分析可知, DHNN 网络存在若干个吸引子. 如果将网络所有的吸引子看作是记忆模式的集合, 而将网络初态看作一个提示模式 (即发生某些变形或含有噪声的记忆模式), 那么, 网络的收敛过程就可以看作一种联想记忆过程. 我们的希望是从一个提示模式下回忆出一个记忆模式, 即从网络的初态收敛到其对应的模式. DHNN 网络吸引子的个数是网络记忆的一种测度, 即记忆容量. 如前所述, DHNN 网络的记忆容量与其工作方式 (串行或并行) 及连接权和阈值 $\boldsymbol{W}, \boldsymbol{\theta}$ 紧密相关.

用 DHNN 实现联想记忆需要考虑两个重要问题: 怎样按记忆要求设计一个网络 (即确定网络的 $\boldsymbol{W}, \boldsymbol{\theta}$); 网络设计确定后, 如何分析其记忆容量.

首先讨论 DHNN 网络的 $\boldsymbol{W}, \boldsymbol{\theta}$ 的设计. 假定有 m 个需要记忆的 n 维模式, $\boldsymbol{W}$, $\boldsymbol{\theta}$ 的设计要使得这 m 个记忆模式对应的状态恰好是网络能量函数的 m 个局部极小点. 这是一个相当困难的问题. $\boldsymbol{W}, \boldsymbol{\theta}$ 的设计方法有外积法、伪逆法、正交化设计法等. 下面仅介绍外积法和伪逆法.

设向量 $\boldsymbol{x}^k = (x_1^k, x_2^k, \cdots, x_n^k)^{\mathrm{T}}, k = 1, 2, \cdots, m, x_i^k \in \{1, -1\}$ 是要求网络记忆的 m 个 $(m < n)$ 个模式向量, 它们彼此正交, 即满足

$$(\boldsymbol{x}^i)^{\mathrm{T}}(\boldsymbol{x}^j) = \begin{cases} 0, & i \neq j \\ n, & i = j \end{cases} \tag{5.4.10}$$

并且网络的 n 个节点的阈值均为 0: $\boldsymbol{\theta} = \boldsymbol{0}$, 如果连接权按下式计算

$$\boldsymbol{W} = \sum_{k=1}^{m} \left[\boldsymbol{x}^k (\boldsymbol{x}^k)^{\mathrm{T}} - \boldsymbol{I}\right]$$

即

$$w_{ij} = \begin{cases} \sum\limits_{k=1}^{m} x_i^k x_j^k, & i \neq j \\ 0, & i = j \end{cases} \tag{5.4.11}$$

其中, $\boldsymbol{I}$ 为 $n \times n$ 阶单位矩阵, 则向量 $\boldsymbol{x}^k, k = 1, 2, \cdots, m$ 都是 DHNN 网络的稳定点. 证明如下:

从 $\boldsymbol{x}^k = (x_1^k, x_2^k, \cdots, x_n^k)^{\mathrm{T}}, k = 1, 2, \cdots, m$ 中取任一向量 $\boldsymbol{x}^j$ 作为网络的初始输入, 则有

$$\boldsymbol{W}\boldsymbol{x}^j = \sum_{k=1}^{m} \left[\boldsymbol{x}^k (\boldsymbol{x}^k)^{\mathrm{T}} - \boldsymbol{I}\right] \boldsymbol{x}^j = \left[\boldsymbol{x}^j (\boldsymbol{x}^j)^{\mathrm{T}} - \boldsymbol{I}\right] \boldsymbol{x}^j + \sum_{k=1, k \neq j}^{m} \left[\boldsymbol{x}^k (\boldsymbol{x}^k)^{\mathrm{T}} - \boldsymbol{I}\right] \boldsymbol{x}^j$$

利用正交性式 (5.4.10), 即得

$$\boldsymbol{W}\boldsymbol{x}^j = (n-1)\boldsymbol{x}^j - (m-1)\boldsymbol{x}^j = (n-m)\boldsymbol{x}^j$$

注意到 $(n - m) > 0$ 和 $\boldsymbol{\theta} = \boldsymbol{0}$, 所以网络的输出按照式 (5.4.2) 为

$$\operatorname{sgn}(\boldsymbol{W}\boldsymbol{x}^j) = \boldsymbol{x}^j$$

即 $\boldsymbol{x}^j, j = 1, 2, \cdots, m$ 是满足条件式 (5.4.5) 的 m 个吸引子. 式 (5.4.11) 的连接权矩阵是要求网络记忆的 m 个 $(m < n)$ 模式向量的外积矩阵, 所以构建该权矩阵的方法称为 "外积" 规则.

外积规则要求网络记忆的 m 个 $(m < n)$ 模式向量相互正交, 条件比较苛刻; 伪逆规则只要求模式向量线性独立, 条件较为宽松.

连接权矩阵的伪逆规则如下: 设向量 $\boldsymbol{x}^k, k = 1, 2, \cdots, m$ 是要求网络记忆的 m 个 $(m < n)$ 模式向量, 它们彼此线性独立. 令

$$\boldsymbol{X} = (\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^m) \tag{5.4.12}$$

是 n 行 m 列矩阵, 由于矩阵 $\boldsymbol{X}$ 中各列向量线性独立, 故 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 是满秩矩阵, 存在逆矩阵. 这时权矩阵可由下式求得:

$$\boldsymbol{W} = \boldsymbol{X}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}} = \boldsymbol{X}\boldsymbol{X}^{+} \tag{5.4.13}$$

其中, $\boldsymbol{X}^{+} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}$ 是矩阵 $\boldsymbol{X}$ 的伪逆. 可以简单地验证

$$\boldsymbol{W}\boldsymbol{X} = \boldsymbol{X}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}) = \boldsymbol{X}$$

所以 $\boldsymbol{X}$ 的 m 个列向量都是网络的吸引子.

由此我们给出 n 个神经元的 DHNN 的联想记忆的以下学习算法:

(1) 给定要求网络记忆的 (即需要对输入样本作分类的类别) m 个 $(m < n)$ 模式向量 $\boldsymbol{x}^k = (x_1^k, x_2^k, \cdots, x_n^k)^{\mathrm{T}}, k = 1, 2, \cdots, m, x_i^k \in \{1, -1\}$, 它们彼此正交或线性独立.

(2) 按照式 (5.4.11) 或式 (5.4.13) 计算连接权矩阵元.

(3) 输入一个样本的特征向量值 $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}, x_i \in \{1, -1\}$ 作为网络初始状态向量 $\boldsymbol{x}(t=0) = [x_1(0), x_2(0), \cdots, x_n(0)]^{\mathrm{T}}$.

(4) 迭代计算: 按式 (5.4.2) 所示的 DHNN 的串行工作方式改变网络状态向量

$$\begin{cases} x_i(t+1) = \mathrm{sgn}\left[\displaystyle\sum_{j=1}^{n} w_{ij}x_j(t)\right] \\ x_j(t+1) = x_j(t), \quad j \neq i \end{cases}$$

直到网络状态向量不再改变稳定为止. 此时的网络状态向量即是输入样本的最佳匹配模式.

下面讨论具有 n 个神经元的 DHNN 的记忆容量问题.

所谓记忆容量, 是指给定网络节点数 n, 网络记忆的模式类别 m 的最大值. 影响记忆容量的因素有:

(a) 网络节点数 n.

(b) 网络记忆的模式向量的性质. 正交的模式向量情形下有最大的记忆容量.

(c) 连接权的设计. 适当的连接权设计可以提高记忆容量.

(d) 吸引子吸引域的大小. 要求吸引域越大, 记忆容量越小.

记忆容量的严格分析是相当困难的. Hopfield 给出了一个估计, 即

$$m \leqslant 0.15n. \tag{5.4.14}$$

按照样本为随机分布的假设所作的理论分析表明, 当 $n \to \infty$ 时, 记忆容量为

$$m \leqslant \frac{(1-2\alpha)^2 n}{2\ln n} \tag{5.4.15}$$

其中, α 为要求的吸引域的半径. 一个网络记忆的模式向量 $\boldsymbol{x}^k$ 的吸引域可以看作以该向量为中心的球体, 落在该球体内的向量 $\boldsymbol{x}^s$ 满足

$$d_{\mathrm{H}}(\boldsymbol{x}^k,\boldsymbol{x}^s)=\frac{1}{2}\sum_{i=1}^{n}(1-x_i^k x_i^s)\leqslant\alpha n \tag{5.4.16}$$

其中, $d_{\mathrm{H}}(\boldsymbol{x}^k,\boldsymbol{x}^s)$ 是 $\boldsymbol{x}^k$ 与 $\boldsymbol{x}^s$ 间的汉明距离, 它表示向量 $\boldsymbol{x}^k$ 与 $\boldsymbol{x}^s$ 间不相等的分量的个数. 任何满足式 (5.4.16) 的特征向量 $\boldsymbol{x}^s$ 输入 DHNN, 最终将收敛于吸引子 $\boldsymbol{x}^k$.

5.4.2 连续 Hopfield 网络

连续型 Hopfield 神经网络 (CHNN) 是 J.J. Hopfield 于 1984 年在 DHNN 的基础上提出来的[35]. 它的基本原理与 DHNN 相似. 由于 CHNN 以模拟量作为网络的输入输出量, 各神经元采用并行方式工作, 所以在信息处理的并行性、联想性、实时性、存储分布性和协同性方面比 DHNN 更接近于生物神经网络.

图 5.12 是 Hopfield 动态神经元模型. 图中电阻 R_{i0} 和电容 C_i 并联, 模拟生物神经元的延时特性. 电阻 $R_{ij}(j=1,2,\cdots,n)$ 模拟生物神经元之间的突触特性. 运算放大器模拟生物神经元的非线性特性, 其输入 u_i 和输出 v_i 按 S 型函数变化, 即

$$v_i=f(u_i). \tag{5.4.17}$$

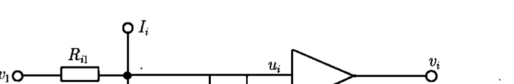
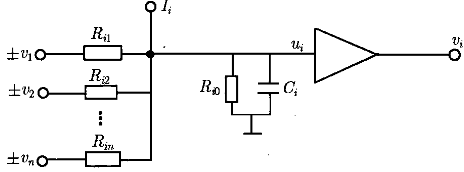



图 5.12 Hopfield 动态神经元模型

其中, f 为 Sigmoid 函数或双曲线正切函数. I_i 为独立的外输入信号.

图 5.13 是 CHNN 的结构图. 对于每一个神经元而言, 自身的输出信号经过其他神经元又反馈到自己, 所以 CHNN 是一个连续的非线性动力学系统. 各放大器输出的反馈权值 w_{ij} 反映神经元之间的突触特性, 但其中不直接反馈回自身, 即自反馈权值为 0: $w_{ii}=0$. 这一点与 DHNN 相同.

对于第 i 个神经元, 放大器的输入输出关系可用下式描述

$$c_i\frac{\mathrm{d}u_i}{\mathrm{d}t}=-\frac{u_i}{R_{i0}}+\sum_{j=1}^{n}\frac{1}{R_{ij}}(v_j-u_i)+I_i \tag{5.4.18}$$

它可改写为

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -\frac{u_i}{\tau_i} + \sum_{j=1}^{n} w_{ij} v_j + \theta_i \tag{5.4.19}$$

其中

$$\begin{cases} \dfrac{1}{\tau_i} = \dfrac{1}{R_{i0}c_i} + \displaystyle\sum_{j=1}^{n} \dfrac{1}{R_{ij}c_i} \\ w_{ij} = \dfrac{1}{R_{ij}c_i} \\ \theta_i = I_i / c_i \end{cases} \tag{5.4.20}$$

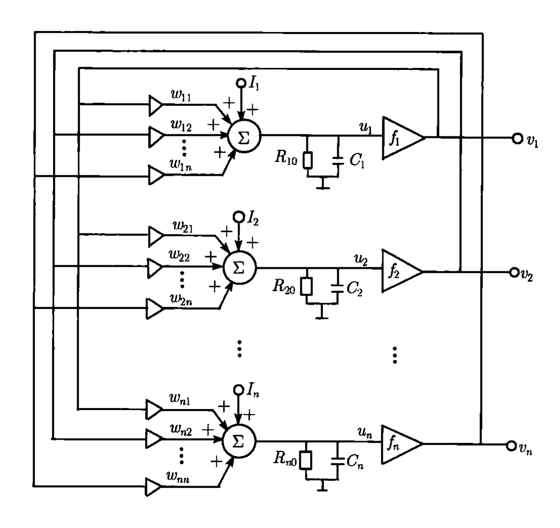


图 5.13　CHNN 结构图

由于 f 为连续函数, 网络中的所有节点的状态随着时间并行地更新, 在一定范围内连续变化.

由 n 个神经元构成的 CHNN, 各放大器的输入输出关系可用下述方程描述

$$\dot{\boldsymbol{u}} = -\boldsymbol{T}^{-1}\boldsymbol{u} + \boldsymbol{W}\boldsymbol{v} + \boldsymbol{\theta} \tag{5.4.21}$$

其中

$$\boldsymbol{u} = [u_1, u_2, \cdots, u_n]^{\mathrm{T}}$$
$$\boldsymbol{v} = [v_1, v_2, \cdots, v_n]^{\mathrm{T}} = \boldsymbol{f}(\boldsymbol{u})$$
$$\boldsymbol{T} = \mathrm{diag}\,[\tau_1, \tau_2, \cdots, \tau_n]^{\mathrm{T}} = [v_1, v_2, \cdots, v_n]^{\mathrm{T}}$$

$$\begin{aligned}\boldsymbol{\theta}&=[\theta_1,\theta_2,\cdots,\theta_n]^{\mathrm{T}}\\ \boldsymbol{W}&=[w_{ij}]_{n\times n}\end{aligned}\tag{5.4.22}$$

与 DHNN 一样, 网络的稳定性分析基于网络的能量函数. CHNN 的能量函数定义为

$$E=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}w_{ij}v_iv_j-\sum_{i=1}^{n}v_iI_i+\sum_{i=1}^{n}\frac{1}{R_i}\int_0^{v_i}f_i^{-1}(v)\mathrm{d}v\tag{5.4.23}$$

式中第三项表示一种输入状态和输出值关系的能量项. 如果 CHNN 中运算放大器为理想或近似理想放大器, 则上式中第三项的能量项可忽略不计, 此时能量函数可表示为

$$E=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}w_{ij}v_iv_j-\sum_{i=1}^{n}v_iI_i\tag{5.4.24}$$

关于式 (5.4.19) 所描述的 CHNN 网络的稳定性有以下定理 (证明从略): 若 $f_i^{-1}(v)$ 为单调递增的连续函数, 并有 $c_i>0,w_{ij}=w_{ji}$, 则网络状态的变化有

$$\begin{aligned}&\frac{\mathrm{d}E}{\mathrm{d}t}\leqslant 0,\\ &\text{当且仅当}\quad \frac{\mathrm{d}v_i}{\mathrm{d}t}=0\ \text{时},\frac{\mathrm{d}E}{\mathrm{d}t}=0.\end{aligned}\tag{5.4.25}$$

该定理表示 CHNN 网络的状态变化向能量函数减小的方向运动, 并最终收敛于网络的 E 的极小值点, 即网络的稳定平衡点.

关于 CHNN 有如下结论:

(1) 具有良好的收敛性, 即从任意非平衡状态出发, 网络将收敛于某个平衡态.

(2) 具有有限个平衡点.

(3) 如果平衡点是稳定的, 它一定是渐近稳定的.

(4) 渐近稳定的平衡点是其能量函数的局部极小点.

(5) 网络的信息存储表现为神经元之间互连的分布式动态存储.

(6) 网络以非线性、连续时间并行方式处理信息, 其计算时间即网络趋于平衡点的时间.

5.4.3 Hopfield 网络在优化计算中的应用

用神经网络求解最优化问题是神经网络应用的一个重要方面. 用 (连续) Hopfield 网络求解最优化问题的过程可以归纳如下:

(1) 选择一种适当的表示方法, 使得网络的状态与待解问题的变量值对应起来.

(2) 把最优化问题的目标函数转化为网络的能量函数, 使其极小值对应于问题的最佳解.

(3) 由能量函数导出网络的结构, 即根据式 (5.4.24) 求出 $\boldsymbol{W}$ 和 $\boldsymbol{\theta}$.

(4) 按网络的工作方式运行网络, 即按式 (5.4.19) 改变网络的状态, 多次迭代后网络的能量函数收敛于极小值, 此时网络的稳定状态就是待求的变量值.

由于神经网络是并行计算, 其计算时间不随维数的增加发生指数性质的“爆炸”, 因而对于最优化问题的高速计算特别有效. 例如, 1985 年 Hopfield 等利用 900 个神经元构成的网络, 仅用 0.2s 就求得了一个 30 个城市的旅行商问题 (TSP) 的最优解. 用其他方法很难做到这一点. 下面以求 TSP 为例, 说明 Hopfield 网络求解最优化问题的方法.

“旅行商最优路径问题 (TSP)”, 是指有 n 个城市 $\boldsymbol{c} = \{c_1, c_2, \cdots, c_n\}$; 城市 c_i, c_j 间的距离用 $d_{ij} = d_{ji}$ 表示. 要求寻找一条经过每个城市仅一次的路程最短且回到出发地的路径. 对于 TSP 问题, 若采用传统的穷举法, 需要找出全部可能的路径 (当 $n \geqslant 3$ 共有 $n!/2n$ 条), 计算并比较它们的长度, 再确定最佳路径. 这种方法随着城市数 n 的增加, 计算量急剧增加. 用传统的串行计算难以短时间内得到结果. 当用 Hopfield 网络求解, 由于神经网络一定程度上模拟了人脑的“思考”功能, 以及并行计算的特点, 避免了传统方法计算量的指数爆炸.

为简明起见, 假定 n=5. 首先把问题转化为适合于神经网络处理的形式. 我们用所谓的换位矩阵 (permutation matrix) $\boldsymbol{V} = [v]_{n\times n}$ 来表示旅行路径, 例如表 5.1 所示的矩阵 $\boldsymbol{V}$ 的值表示了 n=5 的 TSP 问题的一条有效路径. 矩阵 $\boldsymbol{V}$ 中, 行表示城市, 列表示路径次序. $v_{ij} = 1$ 的元素表示它对应的城市 i (行) 在路径中以次序 j (列) 出现 ; $v_{ij} = 0$ 的元素表示不出现. 如果把矩阵 $\boldsymbol{V}$ 的每个元素对应于神经网络的每个神经元, 则该问题可用 $n^2 = 25$ 神经元构成的 Hopfield 网络求解.

表 5.1 n=5 的 TSP 问题的一条有效路径

城市	次序	1	2	3	4	5
c_1		0	1	0	0	0
c_2		0	0	0	1	0
c_3		1	0	0	0	0
c_4		0	0	0	0	1
c_5		0	0	1	0	0

问题求解的第二步是把问题的目标函数转化为网络的能量函数, 并将问题的求解变量与网络的状态对应起来. 解决这个问题往往是求解过程中最关键、最困难的部分, 需要一定的技巧. 根据问题的要求, 一条有效的路径需满足以下约束条件:

(a) 一个城市只能访问一次, 这等价于矩阵 $\boldsymbol{V}$ 每行中只有一个 1.

(b) 一次只能访问一个城市, 这等价于矩阵 $\boldsymbol{V}$ 每列中只有一个 1.

(c) 总共有 n 个城市, 这等价于矩阵 $\boldsymbol{V}$ 所有元素之和为 n.

(d) 要求路径最短, 这等价于网络能量函数的最小值对应于 TSP 问题的最短路径.

现在讨论网络能量函数的构成.

(1) 对应于约束条件 (a), 考虑到矩阵 $\boldsymbol{V}$ 的任意一行的任意两个相邻元素的乘积等于 0, 所以矩阵 $\boldsymbol{V}$ 的 n 行的所有元素按顺序两两相乘之和也为 0, 即 $\sum_{x=1}^{n}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} v_{xi}v_{xj} = 0$. 将它乘以系数 $A/2$, $\frac{A}{2}\sum_{x=1}^{n}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} v_{xi}v_{xj}$ 作为能量函数的第一项.

(2) 对应于约束条件 (b), 可得能量函数的第二项 $\frac{B}{2}\sum_{i=1}^{n}\sum_{x=1}^{n-1}\sum_{y=x+1}^{n} v_{xi}v_{yi}$.

(3) 对应于约束条件 (c), 矩阵 $\boldsymbol{V}$ 的所有元素之和等于 n, 可得能量函数的第三项 $\frac{C}{2}\left[\sum_{x=1}^{n}\sum_{i=1}^{n} v_{xi} - n\right]^2$. 取平方值是为了符合能量函数的表达形式.

(4) 对应于约束条件 (d), 设任意两城市 x, y 间的距离为 d_{xy}, 访问这 2 个城市有两种途径: $x \to y$ 和 $y \to x$, 相应的距离为 $d_{xy}v_{xi}v_{y,i+1}$ 和 $d_{xy}v_{xi}v_{y,i-1}$. 由前三个约束条件可知, 两项中至少有一项为 0. 顺序访问两城市 x, y 的所有可能途径的长度为 $\sum_{i=1}^{n} d_{xy}v_{xi}(v_{y,i+1}+v_{y,i-1})$. 同样由前三个约束条件可知, 这 n 个求和项中, 最多只能有一项 ($d_{xy}v_{xi}v_{y,i+1}$ 或 $d_{xy}v_{xi}v_{y,i-1}$) 不为 0. 如果 n 个求和项均为 0, 则该路径不是按相邻顺序访问这两个城市的.

n 个城市两两之间所有可能的访问路径的长度可表示为

$$\sum_{x=1}^{n}\sum_{y=1}^{n}\sum_{i=1}^{n} d_{xy}v_{xi}(v_{y,i+1}+v_{y,i-1}).$$

其中数值最小的那条路径就是 TSP 问题的最短路径, 由此得到能量函数的第四项:

$$\frac{D}{2}\sum_{x=1}^{n}\sum_{y=1}^{n}\sum_{i=1}^{n} d_{xy}v_{xi}(v_{y,i+1}+v_{y,i-1}).$$

以上所述的前三项仅当问题的约束条件得到满足时才为 0, 从而保证了所得路径的有效性. 当问题的约束条件得不到满足, 网络的能量函数不可能达到极小, 因此它们称为惩罚项. 第四项对应于问题的目标函数, 其最小值即为最短路径长度. 由此得到网络能量函数的表达式:

$$E = \frac{A}{2}\sum_{x=1}^{n}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} v_{xi}v_{xj} + \frac{B}{2}\sum_{i=1}^{n}\sum_{x=1}^{n-1}\sum_{y=x+1}^{n} v_{xi}v_{yi} + \frac{C}{2}\left[\sum_{x=1}^{n}\sum_{i=1}^{n} v_{xi} - n\right]^2$$

$$+\frac{D}{2}\sum_{x=1}^{n}\sum_{y=1}^{n}\sum_{i=1}^{n}d_{xy}v_{xi}(v_{y,i+1}+v_{y,i-1}). \tag{5.4.26}$$

上式符合网络能量函数的定义. 当 E 达到极小时, 由网络状态 v_{ij} 构成的换位矩阵表达了 TSP 问题的最短路径.

问题求解的第三步是确定网络神经元间的连接权和神经元的阈值. 设网络神经元 (x,i) 与 (y,j) 间的连接权为 $w_{xi,yj}$, 神经元 (x,i) 的阈值为 I_{xi}, 则有

$$\begin{cases} w_{xi,yj}=-A\delta_{xy}(1-\delta_{ij})-B\delta_{ij}(1-\delta_{xy})-C-Dd_{xy}(\delta_{j,i+1}+\delta_{j,i-1}) \\ I_{xi}=Cn \\ \delta_{ij}=\begin{cases}1, & (i=j)\\ 0, & (i\neq j)\end{cases}\end{cases}. \tag{5.4.27}$$

实际上, 将上式代入 Hopfield 能量函数表达式 (5.4.24), 则得 TSP 问题的能量函数表达式 (5.4.26)(只差一常数项 n^2).

问题求解的第四步是将网络神经元间的连接权和神经元的阈值代入网络的运行方程式 (5.4.19), 得到求解 TSP 问题的迭代方程如下:

$$\begin{aligned} c_{xi}\frac{\mathrm{d}u_{xi}}{\mathrm{d}t}=&-\frac{u_{xi}}{R_{xi}}-A\sum_{\substack{j=1\\j\neq i}}^{n}v_{xj}-B\sum_{\substack{y=1\\y\neq x}}^{n}v_{yi}-C\left(\sum_{x=1}^{n}\sum_{y=1}^{n}v_{xy}-n\right)\\ &-D\sum_{y=1}^{n}d_{xy}(v_{y,i+1}+v_{y,i-1})\end{aligned}, \tag{5.4.28}$$

$$v_{xi}=f_{xi}(u_{xi})=\frac{1}{2}\left[1+\tanh\left(\frac{u_{xi}}{u_0}\right)\right]. \tag{5.4.29}$$

其中, f 是双曲线正切函数; u_0 是初值.

根据该运行方程, 网络的具体计算步骤如下:

(1) 初始化: 给定初值 u_0 (如 0.02). 为保证收敛于正确解, 按下式给定网络各神经元的初始状态:

$$u_{xi}=\frac{1}{2}u_0\ln(n-1)+\delta_{u_{xi}}$$

这里 $\delta_{u_{xi}}$ 为 $(-1,+1)$ 区间内的随机数.

(2) 按式 (5.4.29) 求得各神经元的输出

$$v_{xi}(t_0)=\frac{1}{2}\left[1+\tanh\left(\frac{u_{xi}(t_0)}{u_0}\right)\right].$$

(3) 按式 (5.4.28) 求得 $\left.\dfrac{\mathrm{d}u_{xi}}{\mathrm{d}t}\right|_{t=t_0}$

(4) 求下一时刻网络的状态

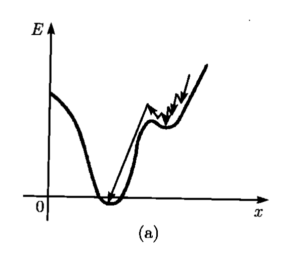
$$u_{xi}(t+\Delta t)=u_{xi}(t)+\left.\frac{\mathrm{d}u_{xi}}{\mathrm{d}t}\right|_{t}\Delta t.$$

(5) 返回步骤 (2), 反复进行运算, 直到网络状态稳定不变. 这时网络的状态对应的换位矩阵 $\boldsymbol{V}$ 的值即为问题的解.

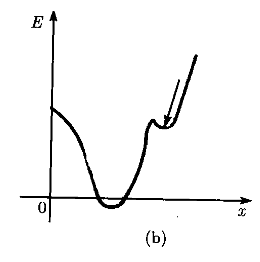
5.5 随机神经网络

5.5.1 随机神经网络的基本思想

神经网络中, 常用某个目标函数的全局极小作为算法搜索和网络状态变化的依据, 如前面讨论的 BP 网络的误差函数和 Hopfield 网络中的能量函数都属于这种情况. 网络的学习或运行过程中其误差函数或能量函数按梯度下降的方向演化, 当梯度趋于 0, 网络的学习或运行过程就停止了. 这种算法往往陷入局部极小点而达不到全局极小点, 被形象地称为 “贪心” 算法 (greedy algorithm), 即急于找到最小解, 结果是欲速则不达. 分析以上两种网络的结构和算法特点, 导致网络陷入局部极小的原因主要有两点: (1) 网络结构存在输入与输出之间的非线性关系, 使网络误差或能量函数所构成的空间是一个包含多个极小的非线性空间. (2) 算法上, 网络误差或能量函数按梯度下降的方向演化而不能有丝毫的上升趋势. 由于第一点为保证网络具有非线性映射能力所必须, 所以解决网络收敛于全局极小的问题只能从第二点着手, 即 “网络误差或能量函数按梯度下降的方向演化” 修改为 “大多数时间按梯度下降的方向演化；某些情况下容许按梯度上升的方向演化”, 则网络就有可能跳出局部极小而向全局极小点收敛. 这就是随机神经网络的基本思想. 图 5.14 是随机算法与贪心算法的形象表示.



(a)



(b)

图 5.14 随机算法与贪心算法的比较

(a) 随机算法; (b) 贪心算法

5.5.2 模拟退火算法

模拟退火算法 (Simulated Annealing Algorithm, 以下简写为 SA 算法) 的基本思想最早是由 Metropolis 于 1953 年针对模拟统计物理中液体结晶问题而提出的一种算法思想[36], 当时是用于模拟物体在给定温度下的热平衡过程. 1983 年 Kirkpatrick 等人把它扩展到温度变化的情况, 并用来求解组合优化问题. SA 算法将组合优化问题与统计物理中的热平衡类比, 开辟了求解组合优化问题的新途径.

SA 算法用于求解组合优化问题是基于固体物质的退火过程与组合优化问题求解过程的类似性. 固体物质的退火处理过程是: 先用高温将它加热熔化, 使其中的粒子可以自由运动. 然后逐渐降低温度, 粒子的自由运动趋势逐渐减弱, 并逐渐形成低能态晶格. 若在凝结点附近温度下降的速度足够慢, 则固体物质一定会形成最低能量的基态, 即最稳定的结构状态. 实际上在整个降温过程中, 各个粒子都可能经历了由高能态向低能态, 有时又暂时由低能态向高能态, 但最终趋向于最低能基态的变化过程. 由此可以得到这样一种启发: 可以把神经网络的状态看作固体内部的 "粒子", 把网络在各个状态下的能量函数看作粒子所处的能态. 在网络的算法中设置一个控制参数 T, 当 T 较大时, 网络能量由低变高的可能性也较大; 随着 T 的减小, 这种可能性也减小. 如果把这个参数看作温度, 使其由高向低慢慢下降, 则整个网络状态的变化过程就完全模拟了固体的退火过程. 当 T 下降到一定程度, 网络将收敛于能量函数的最小值. 可以看到, 网络能量由低变高的可能性是 "网络温度" T 的函数, 用数学模型来表示即网络能量由低变高的概率是 T 的函数.

由此, SA 算法可描述如下. 对于由 n 个神经元组成的反馈网络, 网络的状态用向量 $\boldsymbol{x}^k = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}$ 表示, 各分量是 n 个神经元的输出, 且 x_i 仅取 1 和 0 两个值. 这种情况下, 网络可能有的状态数为 $K = 2^n$, 即 k 的取值为 $k = 1, 2, \cdots, K$. $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_n)^{\mathrm{T}}$ 为网络的阈值向量. $\boldsymbol{W} = [\boldsymbol{W}]_{n\times n}$ 为网络的连接权矩阵, 其元素 w_{ij} 表示神经元 i 和 j 之间的连接权. 权矩阵为对称矩阵, 即有 $w_{ij} = w_{ji}$. 且对角元素为 0, 即 $w_{ii} = 0$. 神经元 i 的综合输入 (即内部状态) 为

$$u_i = \sum_{j=1, j\neq i}^{n} w_{ij} x_j - \theta_i \tag{5.5.1}$$

神经元 i 的输出 x_i 取值 1 和 0 的概率分别为 $p_i(1)$ 和 $p_i(0)$, 它们可表示为

$$p_i(1) = \frac{1}{1 + \mathrm{e}^{-u_i/T}} \tag{5.5.2}$$

$$p_i(0) = 1 - p_i(1) = \frac{\mathrm{e}^{-u_i/T}}{1 + \mathrm{e}^{-u_i/T}} \tag{5.5.3}$$

式中, T 是网络温度. 因此在 SA 算法中, 神经元的输出是由 u_i 为变量的概率 $p_i(1)$ 和 $p_i(0)$ 决定的. 图 5.15 所示为 $p_i(1)$ 的函数曲线.

由式 (5.4.9) 知 Hopfield 网络能量函数的变化为

$$\Delta E_i = -\left[x_i(t+1) - x_i(t)\right] u_i(t).$$

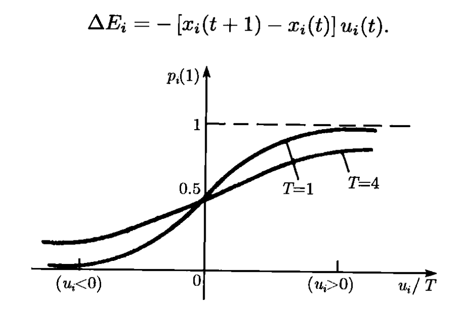


图 5.15 $p_i(1)$ 函数曲线

在 SA 算法中, 当神经元 i 按式 (5.5.2) 的概率, 在下一时刻的输出值取 1 时, 其能量变化为

$$\Delta E_i = -\left[x_i(t+1) - x_i(t)\right] u_i(t). \tag{5.5.4}$$

由上式可以看到当 $u_i(t) \geqslant 0$ 时, $\Delta E_i \leqslant 0$, 表示能量函数随状态的变化是单调减小的; 而当 $u_i(t) < 0$ 时,$\Delta E_i \geqslant 0$, 表示能量函数将增加或不变化. 这在 Hopfield 网络算法中是不容许的, 而在 SA 算法中却容许以比较小的概率 (图 5.15 横轴负值对应的概率) 接受这种变化. 这在有些情况下有利于跳出局部极值. 从图 5.15 还可以看出, 当温度 T 较高时, $p_i(1)$ 相对于 u_i 的变化反应迟钝, 曲线趋于平坦. 特别当 $T \to \infty$ 时, 曲线变为一条恒为 0.5 的直线, 此时 x_i 取值 1 和 0 的概率相等. 这表示当 T 值高时, 网络各神经元有更多的机会进行状态选择, 相当于固体内部的粒子做激烈的自由运动. 当温度降低时, $p_i(1)$ 曲线变陡, $p_i(1)$ 相对于 u_i 的变化相当敏感. 特别当 $T \to 0$ 时, 曲线退化为阶跃函数, SA 算法过渡到离散 Hopfield 网络算法. 所以可以说, 离散 Hopfield 网络算法是 SA 算法 $T \to 0$ 时的特例.

当网络按式 (5.5.1)~(5.5.3) 反复进行状态更新, 且更新次数足够多以后, 可以发现具有能量 E^k 的网络状态 $\boldsymbol{x}^k = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}$ 的出现概率服从 Boltzmann 分布:

$$\begin{cases} P\left(E^k\right) = \dfrac{1}{Z}\mathrm{e}^{-E^k/T} \\ Z = \displaystyle\sum_{k=1}^{K} \mathrm{e}^{-E^k/T} \\ k = 1, 2, \cdots, K, \quad K = 2^n \end{cases} \tag{5.5.5}$$

网络状态 $\boldsymbol{x}^k = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}$ 能量 E^k 的表式为

$$E^k = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} w_{ij}x_i x_j + \sum_{i=1}^{n}\theta_i x_i, \tag{5.5.6}$$

Z 是归一化常数, 等于网络所有 K 种状态的能量和. 由这一分布可以看到, 状态的能量 E^k 越小, 该状态出现的概率越大, 这是 Boltzmann 分布的一大特点, 即能量最小的态以最大的概率出现. 这就保证了 SA 算法收敛于网络的全局极小.

5.5.3 Boltzmann 机及其工作规则

1985 年 Hinton 等人把 SA 算法引入神经网络中[37], 提出了 Boltzmann 机模型, 简称 BM 网络 (Boltzmann machine). BM 网络结构与离散 Hopfield 网络 DHNN 基本相似, 其共同点为:

(1) 每个神经元取二值输出 (如 1 和 0).

(2) 神经元间的连接权矩阵是对称的, 对角元等于 0 (即无自反馈).

(3) 每次只调整一个神经元的状态, 该神经元的抽样是随机的.

不同点是:

(1) BM 网络允许有隐含层 (但没有明显的层次结构), DHNN 则不允许.

(2) BM 网络神经元采用随机激活机制, DHNN 神经元的激活是确定性的.

(3) BM 网络可以以某种随机模式进行有监督的学习, DHNN 在无监督状态下运行.

BM 网络有图 5.16 所示的两种结构. 结构 (a) 由可视层和隐含层两部分组成, 主要用于随机性自联想记忆. 可视层为网络与外界环境提供一个界面. 网络进行训练时, 可视层神经元由外输入向量钳制于特定的状态, 而隐含层神经元则运行在自由状态. 隐含层神经元用于检测外输入的统计特征. 这种网络可以通过无监督学习来模拟外界给定的概率分布.

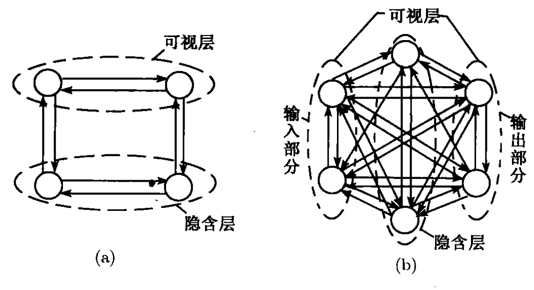


图 5.16 BM 网络的两种结构

结构 (b) 中的可视层进一步分为输入和输出两部分, 它主要用于随机性互联想记忆. 这种网络采用如下的有教师监督的学习方式: 把某个记忆模式加到网络的输入部分, 同时, 网络的输出部分按一定的概率分布给出一组期望输出模式. 此时所给出的概率分布实际上是输出模式相当于输入模式的条件概率分布.

BM 网络的算法根据其两大用途分为工作规则和学习规则. 工作规则也就是网络的状态更新规则, 主要用于求解优化组合问题. 学习规则也就是网络的连接权和阈值的修正规则, 主要用于模拟外界的概率分布.

这里首先介绍 BM 网络的工作规则.

BM 网络的工作规则与 DHNN 工作规则十分相似, 只是以概率方式取代阶跃函数方式对神经元状态进行更新, 而且网络温度随着网络状态的不断更新而逐渐降低. 实际上, BM 网络的工作规则就是模拟退火算法的具体体现. 由于它用于求解优化组合问题, 因此, 它是把问题的原始条件和目标函数转化为网络的能量函数, 按 BM 网络的工作规则进行网络状态的更新, 求得问题的最优解. 在这种情况下, 网络的连接权和阈值应该按联想记忆方式事先设计确定.

对于由 n 个神经元组成的 BM 网络, 网络的状态用向量 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$ 表示, 各分量是 n 个神经元的输出. 网络可能有的状态数为 $K=2^n$. BM 网络工作规则的步骤可归纳如下:

(1) 给定网络阈值 $\boldsymbol{\theta}=(\theta_1,\theta_2,\cdots,\theta_n)^{\mathrm{T}}$, 连接权矩阵 $\boldsymbol{W}=[\boldsymbol{W}]_{n\times n}$ 和初始温度 $T_0\to T(t)$, 输入网络初态 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}\to[x_1(t),x_2(t),\cdots,x_n(t)]^{\mathrm{T}}$. 这时 $t=1$.

(2) 从 n 个神经元中随机选择一个神经元 i, 计算其综合输入, 即内部状态

$$u_i(t)=\sum_{j=1,j\neq i}^{n}w_{ij}x_j(t)-\theta_i$$

(3) 网络状态更新: 神经元 i 之外的神经元状态保持不变

$$x_j(t+1)=x_j(t),\qquad j\neq i,j=1,2,\cdots,n$$

神经元 i 的状态按以下概率进行更新:

$$p_i\left[x_i(t+1)=1\right]=\frac{1}{1+\mathrm{e}^{-u_i(t)/T(t)}}$$

即当 $u_i(t)>0$ 时, $x_i(t+1)=1$;

当 $u_i(t)<0$ 时, $x_i(t+1)$ 的值可有两种方法确定.

$$\text{(a)}\begin{cases}x_i(t+1)=x_i(t) & \text{当}p_i\left[x_i(t+1)=1\right]<r_{0.5}\quad (r_{0.5}\text{为}0\sim0.5\text{间的随机数})\\ x_i(t+1)=1 & \text{其他}\end{cases}$$

(b) $\begin{cases} x_i(t+1) = x_i(t) & \text{当} p_i\,[x_i(t+1)=1] < c_{0.5} \quad (c_{0.5}\text{为}0 \sim 0.5\text{间的常数}) \\ x_i(t+1) = 1 & \text{其他} \end{cases}$

(4) 从 n 个神经元中随机另选一个神经元, 重复步骤 (2)~(3), 直到在温度 $T(t)$ 下网络达到 “热平衡” 状态, 即所有神经元的状态不再变化.

(5) 降低网络温度

$$T(t+1) = \frac{T_0}{\lg(t+1)} \tag{5.5.7}$$

已经证明, 按此降温方案能保证网络收敛于全局极小值, 但它的缺点是收敛速度太慢. 也可用下列快速降温方案:

$$T(t+1) = \frac{T_0}{t+1} \tag{5.5.8}$$

(6) 迭代计算. 令 $t+1 \to t$, 回到步骤 (2) 进行计算, 直到温度 T 小于预先给定的一个截止值 T_{cut}, 迭代结束. 这时, 网络的能量函数 E

$$E = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} w_{ij}x_ix_j + \sum_{i=1}^{n}\theta_i x_i$$

达到极小, 对应的网络状态 $\boldsymbol{x}(T_{\text{cut}}) = (x_1(T_{\text{cut}}), x_2(T_{\text{cut}}), \cdots, x_n(T_{\text{cut}}))^{\mathrm{T}}$ 为待求解问题的最优解.

关于初始温度和结束温度, 目前还没有成熟的设定方法, 一般凭经验给出.

由于 BM 网络的工作规则导致的网络的状态转移, 使得无论从什么初始状态出发, 都收敛到网络能量函数的最小值, 能量函数的各个局部极小值无法被利用来作为模式记忆的存储点, 所以 BM 网络以工作规则运行时, 不能作为多记忆模式的联想记忆器使用.

5.5.4　Boltzmann 机学习规则

网络的学习规则是指网络连接权和阈值的修正规则. BM 网络的学习规则主要通过网络训练模拟外界的概率分布, 实现概率意义上的联想记忆. 所谓概率意义上的联想记忆, 指的是网络所记忆的并不是记忆模式本身, 而是记忆模式出现的概率. 这时, 提供给网络进行训练的也不仅是训练样本, 而且有训练样本出现的概率.

联想记忆可分为自联想记忆和互联想记忆两类. 自联想记忆由图 5.16(a) 所示的 BM 网络实现, 互联想记忆由图 5.16(b) 所示的 BM 网络实现.

1. 自联想记忆学习规则

假定 BM 网络有 N 个神经元, 可视层有 n 个神经元, 隐含层有 m 个神经元, $N = n + m$. 可视层有 $p = 2^n$ 种状态, 隐含层 $q = 2^m$ 种状态, 整个网络有 $K = 2^N = p \cdot q$ 种状态. 可视层状态可表示为 $\boldsymbol{x}_a = (x_1^a, x_2^a, \cdots, x_n^a)^{\mathrm{T}}, a = 1, 2, \cdots, p$;

隐含层状态可表示为 $\boldsymbol{x}_b=(x_1^b,x_2^b,\cdots,x_m^b)^{\mathrm{T}},b=1,2,\cdots,q$. 各分量 x_i 仅取 1 和 0 两个值. 网络的连接权矩阵和阈值向量为 $\boldsymbol{W}=[\boldsymbol{W}]_{N\times N}$ 和 $\boldsymbol{\theta}=(\theta_1,\theta_2,\cdots,\theta_N)^{\mathrm{T}}$.

所谓的自联想记忆, 是指给网络的可视层提供一组记忆模式 $\boldsymbol{x}_a=(x_1^a,x_2^a,\cdots,x_n^a)^{\mathrm{T}},a=1,2,\cdots,p$ 及其中每一个记忆模式应出现的概率 (即这组记忆模式的概率分布函数), 让网络按照下面将介绍的学习规则进行学习. 学习结束后得到相应的权矩阵和阈值向量值. 此后网络从任何初始状态出发, 当网络利用学习过程得到的权矩阵和阈值向量按 5.5.3 小节介绍的工作规则进行不断的状态更新, 网络可视层的各种状态将按学习过程中给定的记忆模式的概率分布出现, 即概率大的状态出现的频率高, 概率小的状态出现的频率低. 这样, 网络相当于一个按既定概率分布输出的 "概率发生器", 这就是概率意义上的自联想记忆. 可以看到, 自联想记忆的实质是网络通过学习目标概率分布, 将其记忆并在以后的回想过程中将这一概率分布再现出来. 应当注意的是, 可视层神经元的个数可根据记忆模式的种类确定, 而隐含层神经元的个数目前需凭借经验确定.

BM 网络怎样记忆目标分布函数呢? 前面已经指出, BM 网络按工作规则进行网络状态更新, 当更新次数足够多, 网络状态出现的概率服从 Boltzmann 分布. Boltzmann 分布函数是由网络状态的能量函数决定的, 而能量函数又是由网络的连接权和阈值所决定. 因此, 通过连接权和阈值的适当调整, 就可实现所期望的 Boltzmann 概率分布. 连接权和阈值的调整过程也就是网络的学习过程.

根据式 (5.5.5) 给出的 Boltzmann 概率分布, 网络的状态概率分布函数 $Q(\boldsymbol{x}_a,\boldsymbol{x}_b)$ 为

$$\begin{cases} Q(\boldsymbol{x}_a,\boldsymbol{x}_b)=\dfrac{1}{Z}\mathrm{e}^{-E_k(\boldsymbol{x}_a,\boldsymbol{x}_b)/T} \\ Z=\displaystyle\sum_{k=1}^{K}\mathrm{e}^{-E_k(\boldsymbol{x}_a,\boldsymbol{x}_b)/T} \\ k=1,2,\cdots,K, \qquad K=2^n \end{cases} \tag{5.5.9}$$

式中, $E_k(\boldsymbol{x}_a,\boldsymbol{x}_b)$ 为网络在状态 k 时 (可视层和隐含层状态分别用 $\boldsymbol{x}_a=(x_1^a,x_2^a,\cdots,x_n^a)^{\mathrm{T}}$ 和 $\boldsymbol{x}_b=(x_1^b,x_2^b,\cdots,x_m^b)^{\mathrm{T}}$ 表示) 的能量函数

$$E_k(\boldsymbol{x}_a,\boldsymbol{x}_b)=-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}w_{ij}x_i^kx_j^k+\sum_{i=1}^{N}\theta_ix_i^k \tag{5.5.10}$$

这时, 可视层实际输出状态的概率分布 $Q(\boldsymbol{x}_a)$ 为

$$Q(\boldsymbol{x}_a)=\sum_{b=1}^{q}Q(\boldsymbol{x}_a,\boldsymbol{x}_b), \qquad a=1,2,\cdots,p \tag{5.5.11}$$

令记忆模式 $\boldsymbol{x}_a=(x_1^a,x_2^a,\cdots,x_n^a)^{\mathrm{T}},a=1,2,\cdots,p$ 的目标概率分布为 $P(\boldsymbol{x}_a)$, $\boldsymbol{x}_a$ 及 $P(\boldsymbol{x}_a)$ 是事先给定的已知值. 为表示目标概率分布 $P(\boldsymbol{x}_a)$ 与实际概率分布 $Q(\boldsymbol{x}_a)$

的偏差, 引用统计学中的 Kullback 偏差 G (也称交叉熵) 的定义

$$G(w_{ij}) = \sum_{a=1}^{p} P(\boldsymbol{x}_a) \cdot \ln \frac{P(\boldsymbol{x}_a)}{Q(\boldsymbol{x}_a)}. \tag{5.5.12}$$

交叉熵具有性质 $G(w_{ij}) \geqslant 0$, 且仅当 $P(\boldsymbol{x}_a) = Q(\boldsymbol{x}_a)$ 时 $G(w_{ij}) = 0$. 显然, $G(w_{ij})$ 越小, 实际输出状态的概率分布 $Q(\boldsymbol{x}_a)$ 就越接近于目标概率分布 $P(\boldsymbol{x}_a)$. 因此网络的学习过程也就是求 $G(w_{ij})$ 极小值的过程.

对应于 w_{ij} 的微小变化 Δw_{ij},$G(w_{ij})$ 的变化量为

$$G(w_{ij} + \Delta w_{ij}) = G(w_{ij}) + \Delta w_{ij} \frac{\partial G(w_{ij})}{\partial w_{ij}}. \tag{5.5.13}$$

如果设

$$\Delta w_{ij} = -\varepsilon \cdot \frac{\partial G(w_{ij})}{\partial w_{ij}}, \qquad \varepsilon > 0. \tag{5.5.14}$$

则必有

$$G(w_{ij} + \Delta w_{ij}) \leqslant G(w_{ij}). \tag{5.5.15}$$

式 (5.5.15) 说明, 如果按式 (5.5.14) 调整连接权, 则网络的交叉熵 $G(w_{ij})$ 呈单调下降趋势. 随着连接权调整的反复进行, $G(w_{ij})$ 将收敛于极小值, 即可实现目标概率分布 $P(\boldsymbol{x}_a)$.

式 (5.5.14) 中 $G(w_{ij})$ 对 w_{ij} 的偏微分可表为

$$\frac{\partial G(w_{ij})}{\partial w_{ij}} = -\frac{1}{T}\left(P_{ij}^{(+)} - P_{ij}^{(-)}\right). \tag{5.5.16}$$

其中

$$P_{ij}^{(+)} = \sum_{a=1}^{p} P(\boldsymbol{x}_a) \cdot \frac{\sum\limits_{b=1}^{q} x_i x_j \mathrm{e}^{-E_k(\boldsymbol{x}_a, \boldsymbol{x}_b)/T}}{\sum\limits_{b=1}^{q} \mathrm{e}^{-E_k(\boldsymbol{x}_a, \boldsymbol{x}_b)/T}}. \tag{5.5.17}$$

$$P_{ij}^{(-)} = \frac{1}{Z} \sum_{k=1}^{K} x_i x_j \mathrm{e}^{-E_k(x_a, x_b)/T}. \tag{5.5.18}$$

$P_{ij}^{(+)}$ 表示网络可视层各神经元输出固定于目标概率分布 $P(\boldsymbol{x}_a)$, 而隐含层各神经元按 Boltzmann 机工作规则进行状态更新足够多次达到平衡状态后, 神经元 i 和 j 同时输出为 1 的概率, 它也称为 x_i 与 x_j 之间的对称概率. $P_{ij}^{(-)}$ 表示网络所有神经元按 Boltzmann 机工作规则进行状态更新足够多次达到平衡状态后, 神经元 i 和 j

同时输出为 1 的概率. 把式 (5.5.17)~(5.5.18) 代入式 (5.5.16) 即得网络连接权的修正值

$$\Delta w_{ij} = \frac{\varepsilon}{T}\left(P_{ij}^{(+)} - P_{ij}^{(-)}\right), \qquad i,j = 1,2,\cdots,N; i \neq j. \tag{5.5.19}$$

该式表示了 BM 网络的自联想记忆的学习方法. 式中第一项表示连接权的调整量 Δw_{ij} 与 $P_{ij}^{(+)}$ 成比例增加, 而 $P_{ij}^{(+)}$ 是神经元 i 与 j 之间的对称概率, 即 x_i 与 x_j 同时为 1 的数量越多, $P_{ij}^{(+)}$ 越大；反之亦然. 这类似于 Hebb 学习原理: 两个神经元同时兴奋, 则它们之间的连接权得以增强. 式中第二项表示连接权的调整量 Δw_{ij} 与 $P_{ij}^{(-)}$ 成比例减小, 这与 Hebb 学习原理正好相反, 因此这一项称为反学习项. 故而, BM 网络通过可视层与 "外界环境" 接触时进行 Hebb 学习；当与 "外界环境" 隔绝时进行反学习.

按照上述讨论, BM 网络自联想记忆学习规则的步骤可归结如下.

(1) 设定常数和初值

N, n(网络神经元数和可视层神经元数)

T_0, T_{E}(初始和结束温度),

ε(学习率)

M, L ($M > 2^n$ 循环次数和状态更新次数)；

记忆模式 $\boldsymbol{x}_a = (x_1^a, x_2^a, \cdots, x_n^a)^{\mathrm{T}}; a = 1,2,\cdots,p; p = 2^n$ 及其目标概率分布 $P(\boldsymbol{x}_a)$；

w_{ij} 赋予 $[-1,+1]$ 区间内的随机值, 并满足 $w_{ij} = w_{ji}, w_{ii} = 0, i,j = 1,2,\cdots,N$, 即无自反馈对称网络, 阈值设为 0: $\theta_i = 0$.

(2) 按给定的目标概率分布 $P(\boldsymbol{x}_a)$ 随机地选取一个模式状态 $\boldsymbol{x}_a$, 将网络可视层各神经元的输出固定在该模式状态 $\boldsymbol{x}_a = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}$.

(3) 从温度 T_0 开始, 按网络工作规则 (模拟退火算法) 对隐含层各神经元的输出进行状态更新, 直至达到 T_{E} 温度下的平衡态 $\boldsymbol{x}_b = (x_1, x_2, \cdots, x_m)^{\mathrm{T}}, m = N - n$.

(4) 在 T_{E} 温度下, 进行 L 次网络全部神经元的状态更新, 每次更新后, 累计计算神经元 i 与 j 的输出 x_i 与 x_j 同时为 $1 (i,j = 1,2,\cdots,N)$ 的次数 $n_{ij}^{(+)}$ (Hebb 学习).

(5) 重新从温度 T_0 开始, 按网络工作规则 (模拟退火算法) 对网络全部神经元的输出进行状态更新, 直至达到 T_{E} 温度下的平衡态 $\boldsymbol{x} = (x_1, \cdots, x_n, x_{n+1}, \cdots, x_{n+m})^{\mathrm{T}}$.

(6) 在 T_{E} 温度下, 进行 L 次网络全部神经元的状态更新, 每次更新后, 累计计算神经元 i 与 j 的输出 x_i 与 x_j 同时为 $1 (i,j = 1,2,\cdots,N)$ 的次数 $n_{ij}^{(-)}$ (反学习).

(7) 返回步骤 (2), 对步骤 (2)~(6) 作 M 次循环.

(8) 计算概率 $P_{ij}^{(+)}$ 和 $P_{ij}^{(-)}$

$$\left.\begin{aligned} P_{ij}^{(+)} &= \frac{n_{ij}^{(+)}}{L \cdot M} \\ P_{ij}^{(-)} &= \frac{n_{ij}^{(-)}}{L \cdot M} \end{aligned}\right\}, \quad i,j = 1,2,\cdots,N$$

(9) 修正网络的连接权

$$w_{ij} + \Delta w_{ij} \to w_{ij}$$

$$\Delta w_{ij} = \frac{\varepsilon}{T}\left(P_{ij}^{(+)} - P_{ij}^{(-)}\right), \quad i,j = 1,2,\cdots,N; i \neq j.$$

(10) 返回步骤 (2), 进行多次循环, 直到对所有的 i,j, 连接权的变化 $\Delta w_{ij}(i,j = 1,2,\cdots,N)$ 很小为止, 学习过程结束. 所得到的连接权 w_{ij} 即为学习过程的成果.

学习结束后, 从任何初始状态出发, 利用所得的连接权 w_{ij} 按工作规则进行多次网络状态的转移, 达到平衡态时, 网络可视层各个状态的出现概率将与网络学习时给定的期望概率分布一致.

2. 互联想记忆学习规则

假定 BM 网络有 N 个神经元, 其中可视层的输入部分有 $n_{\rm i}$ 个神经元, 可视层的输出部分有 $n_{\rm o}$ 个神经元, 隐含层有 m 个神经元, $N = n_{\rm i} + n_{\rm o} + m$. 可视层输入部分有 $p_{\rm i} = 2^{n_{\rm i}}$ 种状态, 输出部分有 $p_{\rm o} = 2^{n_{\rm o}}$ 种状态, 隐含层 $q = 2^m$ 种状态, 整个网络有 $K = 2^N = p_{\rm i} \cdot p_{\rm o} \cdot q$ 种状态. 各部分状态可表示为:

可视层输入部分状态可表示为 $\boldsymbol{x}_a = (x_1^a, x_2^a, \cdots, x_n^a)^{\rm T}, a = 1,2,\cdots,p_{\rm i}$;

输出部分状态可表示为 $\boldsymbol{y}_c = (y_1^c, y_2^c, \cdots, y_n^c)^{\rm T}, c = 1,2,\cdots,p_{\rm o}$;

隐含层状态可表示为 $\boldsymbol{x}_b = (x_1^b, x_2^b, \cdots, x_m^b)^{\rm T}, b = 1,2,\cdots,q$.

各分量 x_i 仅取 1 和 0 两个值. 网络的连接权矩阵和阈值向量为 $\boldsymbol{W} = [\boldsymbol{W}]_{N\times N}$ 和 $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_N)^{\rm T}$.

所谓的互联想记忆, 是指给网络的可视层的输入部分提供一组记忆模式 $\boldsymbol{x}_a = (x_1^a, x_2^a, \cdots, x_n^a)^{\rm T}, a = 1,2,\cdots,p_{\rm i}$, 同时给可视层的输出部分按给定的期望概率分布给出一组期望输出模式, 此概率分布实际上是输出模式相对于输入模式的条件概率分布. 用 $P(\boldsymbol{x}_a, \boldsymbol{y}_c) = P(\boldsymbol{x}_a)P(\boldsymbol{y}_c|\boldsymbol{x}_a)$ 表示期望的联合概率分布, 其中 $P(\boldsymbol{y}_c|\boldsymbol{x}_a)$ 为在输入模式为 $\boldsymbol{x}_a$ 的条件下出现输出模式 $\boldsymbol{y}_c, c = 1,2,\cdots,p_{\rm o}$ 的期望条件概率分布. 让网络按照下面将介绍的学习规则进行学习. 学习结束后得到相应的权矩阵和阈值向量值. 学习结束后的网络在进行回想时, 当给网络提供一输入模式 $\boldsymbol{x}_a$ 后, 对网络除输入部分以外的神经元利用学习过程得到的权矩阵和阈值向量按 5.5.3 小节介绍的工作规则进行不断的状态更新. 网络可视层的输出部分的各种状态将按学习过程中给定的记忆模式的条件概率分布出现, 这就是概率意义上的互联想记忆. 因

此, 互联想记忆的实质是网络通过学习目标概率分布, 将其记忆并在以后的回想过程中将这一概率分布再现出来. 应当注意的是, 可视层神经元的个数可根据记忆模式的种类确定, 而隐含层神经元的个数目前需凭借经验确定.

按照上述讨论, BM 网络互联想记忆学习规则的步骤可归结如下.

(1) 设定常数和初值

N, $n_{\rm i}, n_{\rm o}$ (网络神经元数, 可视层输入部分和输出部分神经元数)

$T_0, T_{\rm E}, \varepsilon$(初始和结束温度, 学习率)

L, M_1, M_2(状态更新次数, $M_1 > 2^{n_{\rm o}}, M_2 > 2^{n_{\rm i}}$, 循环次数) ;

w_{ij} 赋予 $[-1,+1]$ 区间内的随机值, 并满足 $w_{ij} = w_{ji}, w_{ii} = 0, i,j = 1,2,\cdots,N$, 即无自反馈对称网络, 阈值设为 0: $\theta_{\rm i} = 0$.

(2) $p = 2^{n_{\rm i}}$ 个记忆模式中随机地选取一个输入模式 $\boldsymbol{x}_a = (x_1, x_2, \cdots, x_{n_{\rm i}})^{\rm T}$ 加到可视层的输入部分:

(3) 按期望的目标条件概率分布 $P(\boldsymbol{y}_c|\boldsymbol{x}_a)$ 随机地选取网络可视层的输出模式 $\boldsymbol{y}_c$, 将可视层输出部分神经元的输出固定在该输出模式状态 $\boldsymbol{y}_c = (y_1, y_2, \cdots, y_{n_{\rm o}})^{\rm T}$.

(4) 从温度 T_0 开始, 按网络工作规则 (模拟退火算法) 对隐含层各神经元的输出进行状态更新, 直至达到 $T_{\rm E}$ 温度下的平衡态 $\boldsymbol{x}_b = (x_1, x_2, \cdots, x_m)^{\rm T}$,$m = N_i - n_{\rm i} - n_{\rm o}$.

(5) 在 $T_{\rm E}$ 温度下, 进行 L 次网络全部神经元的状态更新, 每次更新后, 累计计算神经元 i 与 j 的输出 x_i 与 x_j 同时为 $1(i,j = 1,2,\cdots,N)$ 的次数 $n_{ij}^{(+)}$ (Hebb 学习).

(6) 重新从温度 T_0 开始, 按网络工作规则 (模拟退火算法) 对网络中除可视层输入部分以外的全部神经元进行状态更新, 直至达到 $T_{\rm E}$ 温度下的平衡态.

(7) 在 $T_{\rm E}$ 温度下, 进行 L 次网络全部神经元的状态更新, 每次更新后, 累计计算神经元 i 与 j 的输出 x_i 与 x_j 同时为 $1(i,j = 1,2,\cdots,N)$ 的次数 $n_{ij}^{(-)}$ (反学习).

(8) 返回步骤 (3), 对步骤 (3)~(7) 作 M_1 次循环.

(9) 计算概率 $P_{ij}^{(+)}$ 和 $P_{ij}^{(-)}$

$$\left.\begin{aligned} P_{ij}^{(+)} &= \frac{n_{ij}^{(+)}}{L \cdot M_1} \\ P_{ij}^{(-)} &= \frac{n_{ij}^{(-)}}{L \cdot M_1} \end{aligned}\right\}, \qquad i,j = 1,2,\cdots,N$$

(10) 修正网络的连接权

$$w_{ij} + \Delta w_{ij} \to w_{ij}$$

$$\Delta w_{ij} = \frac{\varepsilon}{T_{\mathrm{E}}}\left(P_{ij}^{(+)} - P_{ij}^{(-)}\right), \qquad i,j = 1,2,\cdots,N; i \neq j.$$

(11) 返回步骤 (2), 随机选取下一个输入模式. 对步骤 (2)~(10) 进行 M_2 次循环.

(12) 返回步骤 (2), 对步骤 (2)~(11) 进行多次循环, 直到连接权的变化 Δw_{ij},i, $j = 1,2,\cdots,N$ 很小为止, 学习过程结束. 所得到的连接权 w_{ij} 即为学习过程的成果.

学习结束后, 从任何初始状态出发, 利用所得的连接权 w_{ij} 按工作规则进行多次网络状态的转移, 达到平衡态时, 网络可视层各个状态的出现概率将与网络学习时给定的期望概率分布一致.

5.5.5 随机神经网络小结

前面关于模拟退火算法及 Boltzmann 机学习和工作规则的介绍中指出, 这一算法可使网络的能量函数收敛于全局最小值, 从而求得问题的最优解. 但实际情况有时并非如此, 所得到的解可能只是近似的最优解, 其原因分析一下这一算法的收敛过程就可以得到答案. 网络的状态随着更新过程的不断进行, 形成一个状态的序列: $\boldsymbol{x}(0), \boldsymbol{x}(1), \cdots, \boldsymbol{x}(k), \cdots$, 接连的两个状态所对应的网络能量不外乎以下三种情况:

$$\begin{aligned} E\left[\boldsymbol{x}(k+1)\right] &> E\left[\boldsymbol{x}(k)\right] \\ E\left[\boldsymbol{x}(k+1)\right] &= E\left[\boldsymbol{x}(k)\right] \ , \\ E\left[\boldsymbol{x}(k+1)\right] &< E\left[\boldsymbol{x}(k)\right] \end{aligned}$$

其中前两种情况的出现概率比较小. 由于算法的这一特点, 使网络在陷入局部极小时有机会跳出来 ; 但同时网络当前状态对应的能量有可能比前一状态大, 当网络初始温度 T_0 不够大、降温过程太快、结束温度 T_{E} 不够小的情况下, 这种可能性更大, 甚至会产生当前解比状态更新过程中的最好解差得多的现象. 这就是为什么有时模拟退火法的结果反不如其他算法好的原因. 针对这种缺点, 提出了一种改进算法 (improved annealing procedure, 简称 IAP 算法), 这里不再详述, 读者可参考有关文献[38].

尽管模拟退火算法存在一些不足, 但它比快速下降的 "贪心" 算法得到最优解的概率高的多, 且这一算法具有很强的通用性, 特别是对复杂性较高、规模较大、对问题的有关知识了解较少的情况, 它具有明显的优越性. 因为它不像其他算法那样, 需要比较多地依赖问题的有关知识来提高算法的性能. 但是, 在 Boltzmann 学习规则中, 包含着其工作规则, 学习与反学习交替进行, 因此, 网络计算量大, 特别是当网络温度下降速度较慢时, 网络收敛过程十分缓慢, 这是制约这种网络算法应用的主要障碍.

5.6 神经网络用于粒子鉴别

5.6.1 用于带电粒子鉴别的特征变量

作为神经网络在粒子物理实验中的应用, 我们来讨论北京谱仪 III (BESIII) 正负电子对撞实验[39] 中的粒子鉴别问题[40]. 如式 (1.2.1) 所示, 该实验中探测器直接测量的粒子只有有限的几种, 即 $\gamma, \mathrm{e}^{\pm}, \mu^{\pm}, \pi^{\pm}, \mathrm{K}^{\pm}, \mathrm{p}, \bar{\mathrm{p}}$. 我们这里所指的粒子鉴别, 特指带电粒子的鉴别, 不包括 γ 光子的鉴别. 粒子鉴别的目的, 即是根据探测器测量到的一根带电径迹的特征参数, 确定该径迹是何种粒子产生的.

BESIII 谱仪许多子探测器都分成桶部和端盖两部分. 为简单起见, 下面的讨论限于桶部的子探测器. 实验用于带电粒子鉴别的特征变量如下.

1. *径迹动量和飞行方向信息*

43 层信号丝构成的漂移室测量带电粒子的飞行轨迹, 根据带电粒子在 BESIII 均匀螺线管 1T 磁场的偏转半径值 R 可确定其动量 p 和飞行方向的极角 θ:

$$p_{\mathrm{t}}(\mathrm{GeV/c}) = 3 \times 10^{-3} B(\mathrm{Tesla}) R(\mathrm{cm})$$
$$p = p_{\mathrm{t}}/|\sin\theta|$$

我们用径迹动量 p 和横动量 p_{t} 等价地表示其动量和飞行方向. 径迹在漂移室中的有效丝层击中数 N_{layer} 与 p 和 p_{t} 一起作为漂移室的特征变量用于粒子鉴别. 漂移室单层信号丝对于径迹位置的测定精度在垂直于正负电子束流的方向为 130μm, 由此使得动量的确定亦有误差, 当动量为 p =1GeV/c时, 其相对误差为 $\sigma_{\mathrm{p}}/p = 0.5\%$.

2. *$\mathrm{d}E/\mathrm{d}x$ 信息*

所谓 $\mathrm{d}E/\mathrm{d}x$ 是指带电粒子在漂移室气体中飞行单位长度后的电离能量损失. 这里用漂移室信号丝的截断平均脉冲幅度 PH 来表示其相对值, 它与 $\mathrm{d}E/\mathrm{d}x$ 只差一个固定的常数因子. 在同样的动量下, 不同粒子的 $\mathrm{d}E/\mathrm{d}x$ 值是不同的, 因此它可以作为鉴别粒子的特征量. BESIII 漂移室的带电粒子归一化脉冲幅度的动量分布如图 5.17 所示, 其中每种粒子的分布均为有一定宽度的带状, 其宽度反映了电离能量损失的统计不确定性和探测器的有限探测能力, 称为 $\mathrm{d}E/\mathrm{d}x$ 分辨, 其数值约为 (6%~7%).

3. *飞行时间计数器信息*

粒子的飞行时间 t_{TOF} 表示粒子在谱仪对撞中心产生飞行到击中飞行时间计数器 (TOF) 的时间间隔. TOF 测到的粒子速度 $\beta_{\mathrm{TOF}}c$ 和质量平方 m^2_{TOF} 可由下式计算:

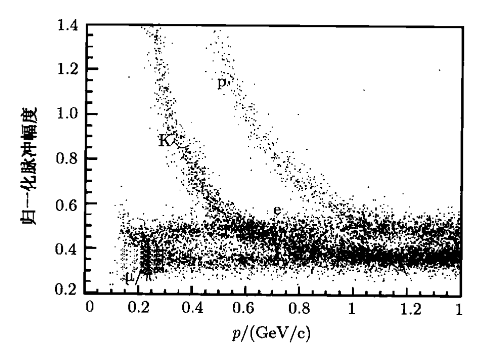


图 5.17　BESIII 漂移室的带电粒子归一化脉冲幅度的动量分布

$$\beta_{\mathrm{TOF}}=\frac{L}{ct_{\mathrm{TOF}}},\qquad m_{\mathrm{TOF}}^2=p^2\frac{1-\beta_{\mathrm{TOF}}^2}{\beta_{\mathrm{TOF}}^2}$$

式中, L 是飞行距离, 由漂移室测得的径迹击中点拟合磁场作用下形成的螺旋线长度求得. 图 5.18 给出了 BESIII 的 TOF 系统对不同粒子的 m_{TOF}^2 随动量的分布. 由于 TOF 对飞行时间 t_{TOF} 的测量存在误差, 因此对同一种粒子, 该分布都是一条带, 带的宽度表征了测量精度, 它由 TOF 的时间分辨 (两层 TOF 的时间分辨测定值为 $\sigma_{\mathrm{TOF}}\leqslant(87.9\pm3.9)\mathrm{ps}$) 决定. 显然, 对于不同的粒子, 其 m_{TOF}^2 是不同的, 因此 m_{TOF}^2 可作为粒子鉴别的特征量. 此外, 径迹击中 TOF 系统的 z 向位置 z_{TOF} 亦作为 TOF 系统提供的粒子鉴别特征量.

4. CsI(Tl) **电磁量能器信息**

由 6272 块 CsI(Tl) 晶体构成的电磁量能器 (简写为 EMC) 可以对光子和带电

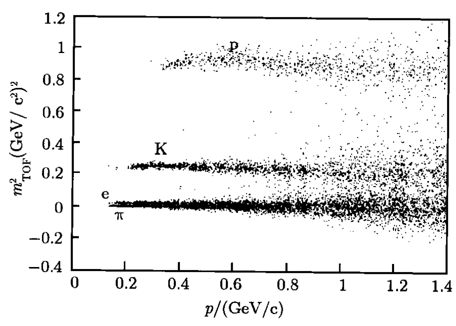


图 5.18　BESIII 的 TOF 系统对不同粒子的 m_{TOF}^2 随动量的分布

粒子在其中的沉积能量进行测量. 不同的带电粒子在 EMC 的沉积能量不同, 不同的带电粒子在 EMC 中簇射形状的不同有助于电子与强子 (π, K, p) 的鉴别和 μ 与强子的鉴别. 簇射的形状可由以下几个特征量来表征:

E_{seed}: 带电粒子击中 EMC, 沉积能量最大的那块中心晶体中的沉积能量.

$E_{3\times3}$: 中心晶体周围 3 × 3 块晶体阵列中的沉积能量和.

$E_{5\times5}$: 中心晶体周围 5 × 5 块晶体阵列中的沉积能量和.

μ_2: 能量沉积的二阶中心矩, 定义为

$$\mu_2 = \frac{\sum_i E_i \cdot d_i}{\sum_i E_i}$$

其中, E_i 是径迹在第 i 块晶体中的沉积能量; d_i 是该晶体与所有晶体沉积能量的重心之间的距离. 图 5.19 是 BESIII 的 EMC 系统对粒子 e, μ, π 鉴别所提供的信息的图示.

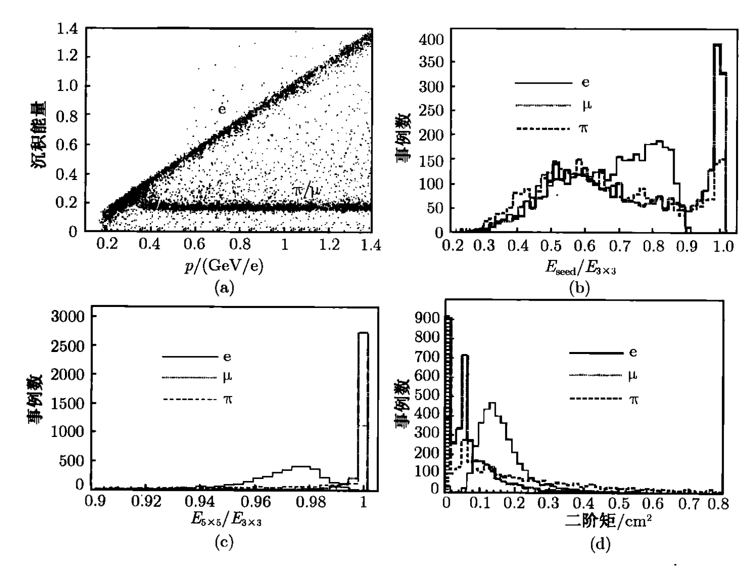


图 5.19 BESIII 的 EMC 系统对粒子 e, μ, π 鉴别提供的信息

(a) 沉积能量的动量分布; (b) $E_{\text{seed}}/E_{3\times3}$ 的分布; (c) $E_{3\times3}/E_{5\times5}$ 的分布; (d) 二阶矩 μ_2 的分布

5. μ 探测器信息

由 9 层阻性板室 (简称 RPC) 以及 8 层轭铁构成的 μ 探测器处于 BESIII 的最外层, 每层阻性板室的平均探测效率 95%, 空间分辨 (即粒子击中点的测量不确定性) 16.6mm.

电子的能量几乎全部被量能器吸收, 不能到达 μ 探测器. 大部分强子穿过量能器后被第一层轭铁吸收；μ 子则有较强的穿透力而被 μ 探测器记录下来. 强子中的 π 有一定的概率能到达 μ 探测器, 但它的贯穿深度 $L_{\rm dep}$ 比 μ 子小. 一般 μ 子在一层阻性板室的读出条上只有一个击中, 而 π 如果在 μ 探测器中发生强子簇射则在一层中可有多次击中. 用 $n_{\mu\rm hit}$ 表示 9 层阻性板室中最大的单层击中数, 因此 $L_{\rm dep}$ 和 $n_{\mu\rm hit}$ 被用作鉴别粒子的特征量. 图 5.20 给出 BESIII 的 μ 探测器系统对粒子 μ, π 鉴别提供的信息.

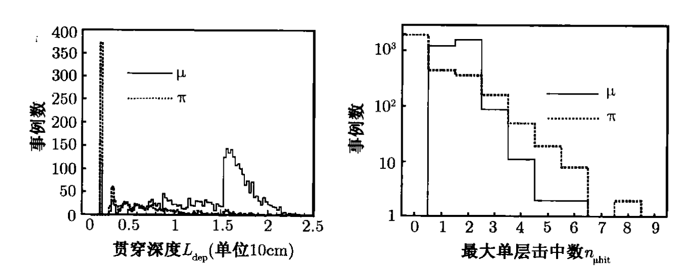


图 5.20　BESIII 的 μ 探测器系统对粒子 μ, π 鉴别提供的信息

5.6.2　带电粒子鉴别的神经网络的架构

为了鉴别 e, μ, π, K, p 五类粒子, 基于物理考虑和网络运行的有效性, 粒子鉴别被分成三个部分：μ 子的判选, 电子的判选以及强子之间的鉴别. BESIII 的粒子鉴别采用了一种新的网络架构, 即首先将各子探测器的信息单独处理, 然后再耦合在一起, 给出被判别粒子的种类. 其优点是降低了网络的规模, 而且避免了不同探测器信息之间虚假关联的产生. 整个网络分为初级和次级两层 (见图 5.21). 初级网络有 4 个子网络 $N_{{\rm d}E/{\rm d}x}$, $N_{\rm TOF}$, $N_{\rm EMC}$, $N_{\rm MUC}$, 编号 1, 2, 3, 4 分别处理 4 个子探测器各自的粒子鉴别信息, 并产生相应的关于粒子种类的输出信息 $O_{\rm dE/dx}$, $O_{\rm TOF}$, $O_{\rm EMC}$, $O_{\rm MUC}$. 这些输出作为次级网络的输入. 次级网络有 9 个子网络 $N_{\rm d}$, $N_{\rm dt}$, $N_{\rm de}$, $N_{\rm dm}$, $N_{\rm dte}$, $N_{\rm dtm}$, $N_{\rm dem}$, $N_{\rm dtem}$, $N_{\rm em}$, 编号 5, 6, 7, 8, 9, 10, 11, 12, 13, 它们关于粒子种类的输出结果表示为 $O_{\rm d}$, $O_{\rm dt}$, $O_{\rm de}$, $O_{\rm dm}$, $O_{\rm dte}$, $O_{\rm dtm}$, $O_{\rm dem}$, $O_{\rm dtem}$, $O_{\rm em}$. 对于 e, μ, π, K, p 五类粒子, 这 13 个子网的期望输出值分别为 1, 2, 3, 4, 5. 但是由于性能的局限, 实际输出对于期望输出存在偏离. 偏离越小, 网络的粒子鉴别性能

越好. 12 个子网都是包含一个隐含层的前馈网络, 采用误差逆传播算法, 激活函数采用 Sigmoid 函数. 唯一的例外是 5 号子网, 它是没有隐含层的单层网络. 13 个子网的有关参数见表 5.2.

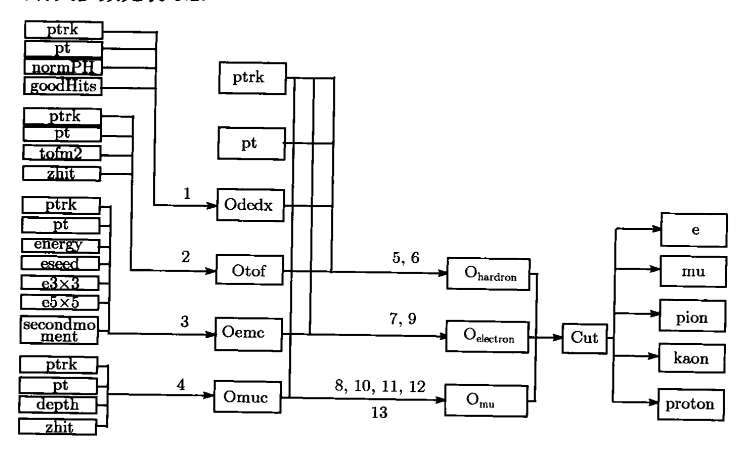


图 5.21 BESIII 的粒子鉴别网络的架构

表 5.2 BESIII 粒子鉴别神经网络的 13 个子网的参数

(名称带有波浪下划线的子网被最终用于粒子鉴别.)

编号	子网名称	训练样本粒子种类	输入特征变量	隐含层神经元数目
1	$\underline{N_{\mathrm{d}E/\mathrm{d}x}}$	e, μ, π, K, p	$p, p_{\mathrm{t}}, PH, N_{\mathrm{layer}}$	20
2	$\underline{N_{\mathrm{TOF}}}$	e, π, K, p	$p, p_{\mathrm{t}}, m^2_{\mathrm{TOF}}, z_{\mathrm{TOF}}$	8
3	$\underline{N_{\mathrm{EMC}}}$	e, μ, π	$p, p_{\mathrm{t}}, E_{\mathrm{seed}}, E_{3\times3}, E_{5\times5}, \mu_2$	10
4	$\underline{N_{\mathrm{MUC}}}$	μ, π	$p, p_{\mathrm{t}}, L_{\mathrm{dep}}, n_{\mu\mathrm{hit}}$	8
5	N_{d}	π, K, p	$O_{\mathrm{d}E/\mathrm{d}x}$	0
6	$\underline{N_{\mathrm{dt}}}$	π, K, p	$p, p_{\mathrm{t}}, O_{\mathrm{d}E/\mathrm{d}x}, O_{\mathrm{TOF}}$	8
7	N_{de}	e, μ, π	$p, p_{\mathrm{t}}, O_{\mathrm{d}E/\mathrm{d}x}, O_{\mathrm{EMC}}$	8
8	N_{dm}	μ, π	$p, p_{\mathrm{t}}, O_{\mathrm{d}E/\mathrm{d}x}, O_{\mathrm{MUC}}$	8
9	$\underline{N_{\mathrm{dt}}}$	e, μ, π	$p, p_{\mathrm{t}}, O_{\mathrm{d}E/\mathrm{d}x}, O_{\mathrm{TOF}}, O_{\mathrm{EMC}}$	10
10	N_{dtm}	μ, π	$p, p_{\mathrm{t}}, O_{\mathrm{d}E/\mathrm{d}x}, O_{\mathrm{TOF}}, O_{\mathrm{MUC}}$	10
11	N_{dem}	μ, π	$p, p_{\mathrm{t}}, O_{\mathrm{d}E/\mathrm{d}x}, O_{\mathrm{TOF}}, O_{\mathrm{MUC}}$	10
12	N_{dtem}	μ, π	$p, p_{\mathrm{t}}, O_{\mathrm{d}E/\mathrm{d}x}, O_{\mathrm{TOF}}, O_{\mathrm{EMC}}, O_{\mathrm{MUC}}$	12
13	$\underline{N_{\mathrm{em}}}$	μ, π	$p, p_{\mathrm{t}}, O_{\mathrm{EMC}}, O_{\mathrm{MUC}}$	8

进一步, 子网络 8, 10~13 用来作为 μ 子的判选, 子网络 7, 9 用作电子的判选, 子网络 5, 6 用作强子之间的鉴别. 后面将会讲到, 经过测试, 最终是用粒子鉴别性

能最优的子网络 13, 9, 6 实行 μ 子、电子的判选和强子之间的鉴别, 它们的输出作为整个网络关于粒子种类的结果 $O_{\mathrm{mu}}, O_{\mathrm{electron}}$ 和 O_{hadron}. 这些输出值经过最后的判选被确定为 5 类粒子, 即输出值在 (0.1~1.4) 之间判为电子, 输出值在 (1.8~2.3) 之间判为 μ 子, (2.5~3.5) 之间判为 π, (3.5~4.5) 之间判为 K, >4.5 判为质子.

前面已经提到, 12 个子网都是包含一个隐含层的前馈网络, 隐含层神经元个数的确定要考虑诸多因素, 如输入输出神经元的个数, 训练样本的大小, 学习所要逼近的函数复杂程度, 网络的具体架构, 网络算法等等. 采用的准则是在不降低粒子鉴别效果的前提下利用尽可能少的隐含层神经元数目. 通过实际的测试, 采用一个 "$2n$" 规则, 即隐含层神经元数目等于输入层神经元个数 (即输入特征变量个数) 的 2 倍. 除了对子网 1 的隐含层神经元数目作了专门的调整, 其他子网隐含层神经元数目都符合这一规则.

5.6.3　网络的训练和粒子鉴别效果

各子网所用的训练样本的粒子种类已经列于表 5.2, 它们是根据各子探测器的鉴别能力和不同粒子在该子探测器中容易混淆的程度而决定的. 子网的训练样本是每种粒子样本量 50000 (区分正、反粒子), 在 (0.1~1.6)GeV/c 动量区间和 $\cos\theta$ (−0.83~0.83) 方向区间内随机地产生均匀分布的单个粒子. 用训练样本确定了 13 个子网各自的连接权和阈值后, 用与训练样本同样数量、同样性质, 但随机数种子不同的检测样本来检测网络的性能.

4 个初级子网的性能见图 5.22. 由图可见, 子网 $N_{\mathrm{d}E/\mathrm{d}x}$ 对于 μ 和 π 几乎没有鉴别能力, 对于 $\mathrm{e}, \mu/\pi, \mathrm{K}, \mathrm{p}$ 有鉴别能力, 但其鉴别能力在不同的动量处有所不同. 在 200MeV 附近 $\mathrm{e}, \mu/\pi$ 混淆在一起, 600MeV 附近 $\mathrm{e}, \mu/\pi, \mathrm{K}$ 混淆在一起, 1200MeV 附近 $\mu/\pi, \mathrm{K}, \mathrm{p}$ 混淆在一起. 子网 N_{TOF} 同样对于 μ 和 π 几乎没有鉴别能力, 但对 600MeV 以下 $\mathrm{e}, \mu/\pi$ 的鉴别有重要贡献, 同时具有很强的 $\mathrm{e}/\mu/\pi, \mathrm{K}, \mathrm{p}$ 鉴别能力, 特别对于质子, 其输出值非常接近于期望值 5. 子网 N_{EMC} 对于 400MeV 以上的 μ 和 π 具有较好的分辨能力. 对于 300MeV 的电子, 其输出值非常接近于期望值 1, 可以与 μ 和 π 清晰地区分开来. 子网 N_{MUC} 对于 500MeV 以上的 μ 和 π 具有较好的分辨能力. 其下界 500MeV 是由于只有动量高于此值的 μ 子才能穿透 μ 子探测器前面的物质.

对于次级子网的检测结果表明, 作为 μ 子判选的子网络 8, 10~13 中, 子网络 13 即 N_{em} 性能最优. 判选电子的子网络 7, 9 中, 子网络 9 即 N_{dte} 性能较好. 用作强子之间鉴别的子网络 5, 6 中则选用鉴别能力强的 6 号子网 N_{dt}. 这 3 个子网的鉴别能力见图 5.23. 我们注意到, 这 3 个子网对于 $\mathrm{e}, \mu, \pi, \mathrm{K}, \mathrm{p}$ 五种粒子的输出值随着动量的变化比 4 个初级子网要平稳得多, 而且相当接近它们的期望值 1, 2, 3, 4, 5. 这是由于综合了各子网的粒子鉴别能力后, 大大提高了整个网络的粒子鉴别的

正确性和稳定性.

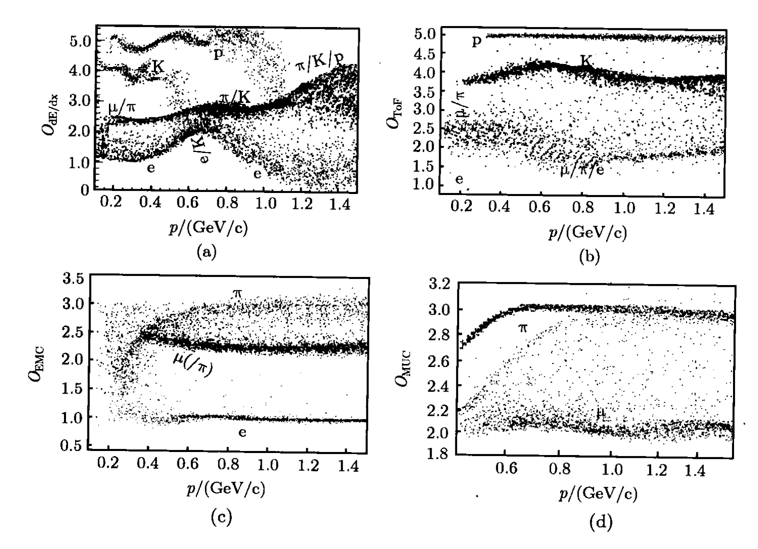


图 5.22 4 个初级子网的粒子鉴别性能

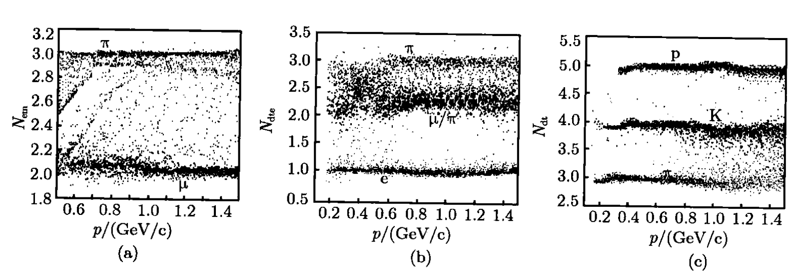


图 5.23 3 个次级子网 N_{em}, N_{dte}, N_{dt} 的粒子鉴别性能

最后次级网络的输出作为整个网络关于粒子种类的输出值, 用 5 类粒子的检测样本确定了粒子种类的判据为: 输出值在 (0.1~1.4) 之间判为电子, 输出值在 (1.8~2.3) 之间判为 μ 子, (2.5~3.5) 之间判为 π, (3.5~4.5) 之间判为 K, >4.5 判为质子. 依照这样的判据, 网络对于 e, μ, π, K, p 五种粒子的判选效率和误判率如图 5.24 和 5.25 所示. 由图可见当动量高于 800MeV, μ 子的判选效率约 90%, 来自 π 的污染率约 5%并随动量的增加而减小, 来自 K 的污染率随动量的增加而增大.

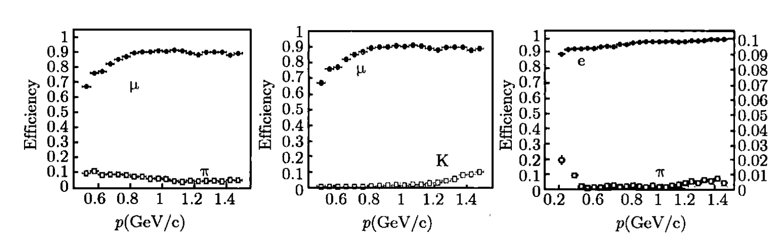


图 5.24　BESIII 粒子鉴别网络对 μ 和 e 的判选效率和误判率

(其中最右边的图中, π 的误判率的纵坐标单位由图右的数字给定).

对于电子的判选从动量 200MeV 开始效率即达 90%, 并随动量的增加而增大. 在动量 (0.25~1.5)GeV 范围内, 来自 π 的污染率小于 1%. 对于强子的鉴别中, 质子的判选效率在整个动量范围内接近 100%, 来自 π 和 K 的污染率很小；π 和 K 在低动量端 (<0.9GeV) 有相当高的判选效率, 相互之间的污染率比较低；但随着动量的增加, 效率逐渐降低而相互间的污染率增大.

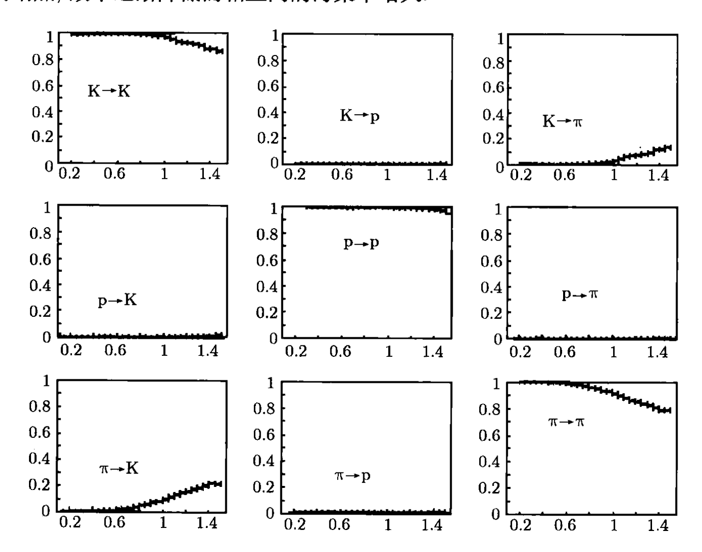


图 5.25　BESIII 粒子鉴别网络对强子的判选效率和误判率

(横坐标为粒子动量 p/(GeV/c))

第六章　近　邻　法

近邻法最初由 Cover 和 Hart[41,42] 于 1967 年提出. 由于诸多研究者对该方法进行了深入的理论研究和发展, 目前已成为模式识别非参数法中的重要方法之一.

6.1　最近邻法

假定有 c 个模式类, 用 $\omega_1, \cdots, \omega_c$ 表示, 有已知类别的训练样本共 N 个, 其中属于 l 类的训练样本为 N_l 个, $l = 1, 2, \cdots, c$, 即 $N = \sum_{l=1}^{c} N_l$. 最近邻法的决策思想是简单而又直观的: 对于任意待归类的样本 $\boldsymbol{x}$, 只要比较 $\boldsymbol{x}$ 与 N 个训练样本之间的欧氏距离, 判定样本 $\boldsymbol{x}$ 与离它距离最近的那个训练样本同类.

我们规定属于 ω_l 类的判别函数为

$$g_l(\boldsymbol{x}) = \min_i d(\boldsymbol{x}, \boldsymbol{x}_i^l), \qquad i = 1, 2, \cdots, N_l, \tag{6.1.1}$$

式中, $\boldsymbol{x}_i^l$ 的角标 l 表示 ω_l 类; i 表示 ω_l 类 N_l 个训练样本中的第 i 个样本. $g_l(\boldsymbol{x})$ 表示模式类 ω_l 的 N_l 个样本中最靠近样本 $\boldsymbol{x}$ 的那个样本与 $\boldsymbol{x}$ 之间的欧氏距离. 于是最近邻法的决策规则可写为, 对任意样本 $\boldsymbol{x}$

若 $g_{\mathrm{m}}(\boldsymbol{x}) = \min_l g_l(\boldsymbol{x})$, $l = 1, 2, \cdots, c$, 则决策

$$\boldsymbol{x} \in \omega_{\mathrm{m}}. \tag{6.1.2}$$

其中, $g_m(\boldsymbol{x})$ 表示 $g_l(\boldsymbol{x})$, $l = 1, 2, \cdots, c$ 中的最小值.

判定样本 $\boldsymbol{x}$ 与离它距离最近的那个训练样本同类, 显然容易导致对样本 $\boldsymbol{x}$ 类别的误判, 因此, 必须讨论最近邻法的错误率问题. 设 N 个样本下的平均错误率为 $\varepsilon_N(e)$, 且样本 $\boldsymbol{x}$ 的最近邻训练样本为 $\boldsymbol{x}'$. 我们注意到, 当对不同的包含 N 个样本的训练样本集应用最近邻法对 $\boldsymbol{x}$ 进行分类时, $\boldsymbol{x}$ 的最近邻训练样本 $\boldsymbol{x}'$ 是不同的, 所以条件错误率与 $\boldsymbol{x}$ 和 $\boldsymbol{x}'$ 都有关, 即应表示为 $\varepsilon_N(e|\boldsymbol{x}, \boldsymbol{x}')$. 若对 $\boldsymbol{x}$ 和 $\boldsymbol{x}'$ 求平均, 则得到 N 个样本下的平均错误率 $\varepsilon_N(e)$:

$$\varepsilon_N(e) = \int_{-\infty}^{\infty} \varepsilon_N(e|\boldsymbol{x}, \boldsymbol{x}') p(\boldsymbol{x}'|\boldsymbol{x}) \mathrm{d}\boldsymbol{x}' p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \tag{6.1.3}$$

定义最近邻法的渐近平均错误率 ε 为 $N \to \infty$ 时 $\varepsilon_N(e)$ 的极限, 记为

$$\varepsilon = \lim_{N \to \infty} \varepsilon_N(e) \tag{6.1.4}$$

可以证明存在下述关系 (证明从略)

$$\varepsilon_{\mathrm{B}} \leqslant \varepsilon \leqslant \varepsilon_{\mathrm{B}}\left(2-\frac{c}{c-1}\varepsilon_{\mathrm{B}}\right) \tag{6.1.5}$$

其中, ε_{B} 为贝叶斯决策下的平均错误率. 由式 (2.1.12) 知

$$\varepsilon_{\mathrm{B}}=\int_{-\infty}^{\infty}\varepsilon_{\mathrm{B}}(e,\boldsymbol{x})\mathrm{d}\boldsymbol{x}=\int_{-\infty}^{\infty}\varepsilon_{\mathrm{B}}(e|\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{6.1.6}$$

式中, $\varepsilon_{\mathrm{B}}(e|\boldsymbol{x})$ 为样本 $\boldsymbol{x}$ 在贝叶斯决策下的条件错误概率; $p(\boldsymbol{x})$ 为随机变量 $\boldsymbol{x}$ 的边沿概率. 对于 c 类问题, $p(\boldsymbol{x})$ 的表式为

$$p(\boldsymbol{x})=\sum_{i=1}^{c}p(\boldsymbol{x}|\omega_i)\pi(\omega_i). \tag{6.1.7}$$

式中, $\pi(\omega_i), i=1,\cdots,c$ 为模式类 $\omega_1,\cdots,\omega_c$ 的先验概率; $p(\boldsymbol{x}|\omega_i)$ 为 $\boldsymbol{x}\in\omega_i$ 时的条件概率密度. 式 (6.1.5) 给出了最近邻法的渐近平均错误率 ε 的范围. 图 6.1 显示了最近邻法的渐近平均错误率 ε 的上、下界与贝叶斯决策下的平均错误率 ε_{B} 之间的关系. ε_{B} 可为 $0\sim(c-1)/c$ 之间的某个值, 最近邻法的渐近平均错误率 ε 落在图中的阴影区域中. 当 $\varepsilon_{\mathrm{B}}=0$ 和 $\varepsilon_{\mathrm{B}}=(c-1)/c$ 时有 $\varepsilon=\varepsilon_{\mathrm{B}}$; 其他情况下则 $\varepsilon>\varepsilon_{\mathrm{B}}$.

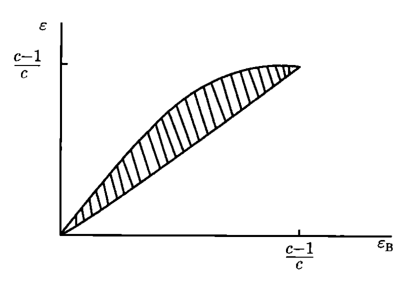


图 6.1　最近邻法错误率 ε 的上下界与贝叶斯决策错误率 ε_{B} 之间的关系

6.2　k 近邻法

k 近邻法是最近邻法的一种推广. 它的基本思想如下: 对于任意待归类的样本 $\boldsymbol{x}$, 取它的 k 个近邻训练样本, 这 k 个近邻样本中哪一个模式类的样本数量最多, 就把样本 $\boldsymbol{x}$ 判为哪一类.

具体来说, 假定有 c 个模式类, 用 $\omega_1,\cdots,\omega_c$ 表示, 有已知类别的训练样本共 N 个, 其中属于 l 类的训练样本为 N_l 个 $(l=1,2,\cdots,c)$. 在这 N 个样本中, 找出待归

类样本 $\boldsymbol{x}$ 的 k 个近邻样本, 其中属于 $\omega_1,\cdots,\omega_c$ 类的样本数为 $k_1,\cdots,k_c$ 个. 我们定义判别函数为

$$g_l(\boldsymbol{x})=k_l,\qquad l=1,2,\cdots,c. \tag{6.2.1}$$

k-近邻法的决策规则可写为, 对任意样本 $\boldsymbol{x}$

若 $g_{\mathrm{m}}(\boldsymbol{x})=\max\limits_{l} k_l,\ l=1,2,\cdots,c$, 则决策

$$\boldsymbol{x}\in\omega_{\mathrm{m}}. \tag{6.2.2}$$

直观地可以判断, 由于利用了未知样本 $\boldsymbol{x}$ 的 k 个近邻训练样本的信息来判断 $\boldsymbol{x}$ 的类别, k 近邻法的平均错误率应当低于最近邻法. 这里我们不加证明地给出对于两类问题 k 近邻法平均错误率 ε_k 的上、下界的表达式. 对于两类问题, 样本 $\boldsymbol{x}$ 在贝叶斯决策下的条件错误概率为

$$\varepsilon_{\mathrm{B}}(e|\boldsymbol{x})=\min\left[q(\omega_1|\boldsymbol{x}),q(\omega_2|\boldsymbol{x})\right] \tag{6.2.3}$$

其中, $q(\omega_i|\boldsymbol{x})$ 为贝叶斯后验概率. 当 $N\to\infty$ 时 k 近邻法的渐近条件错误率 $\varepsilon_k^{N\to\infty}(e|\boldsymbol{x})$ 可表示为

$$\varepsilon_k^{N\to\infty}(e|\boldsymbol{x})\leqslant u_k\left[\varepsilon_{\mathrm{B}}(e|\boldsymbol{x})\right] \tag{6.2.4}$$

其中, $u_k\left[\varepsilon_{\mathrm{B}}(e|\boldsymbol{x})\right]$ 为大于 $\varepsilon_k^{N\to\infty}(e|\boldsymbol{x})$ 的最小凹函数. 这时, 两类问题 k 近邻法平均错误率 ε_k 可由渐近条件错误率 $\varepsilon_k^{N\to\infty}(e|\boldsymbol{x})$ 求平均得到:

$$\varepsilon_k=E\left[\varepsilon_k^{N\to\infty}(e|\boldsymbol{x})\right]\leqslant E\left\{u_k\left[\varepsilon_{\mathrm{B}}(e|\boldsymbol{x})\right]\right\}\leqslant u_k\left\{E\left[\varepsilon_{\mathrm{B}}(e|\boldsymbol{x})\right]\right\}=u_k(\varepsilon_{\mathrm{B}}) \tag{6.2.5}$$

其中, ε_{B} 为贝叶斯决策下的平均错误率 [见式 (6.1.6)]. 于是, 可得到两类问题 k 近邻法平均错误率 ε_k 的上、下界为

$$\varepsilon_{\mathrm{B}}\leqslant\varepsilon_k\leqslant u_k(\varepsilon_{\mathrm{B}})\leqslant u_{k-1}(\varepsilon_{\mathrm{B}})\leqslant\cdots\leqslant u_1(\varepsilon_{\mathrm{B}})\leqslant 2\varepsilon_{\mathrm{B}}\left(1-\varepsilon_{\mathrm{B}}\right) \tag{6.2.6}$$

该式的最后一项即为式 (6.1.5)c=2 的情形, 即两类问题的最近邻法错误率的上限. 由于 $\varepsilon_k^{N\to\infty}(e|\boldsymbol{x})$ 随着 k 的增大单调地减小, 因此式 (6.2.6) 中最小凹函数 u_k 也随着 k 的增大单调地减小. 图 6.2 给出两类情形下 (c=2) k 近邻法错误率 ε_k 的上下界与贝叶斯决策错误率 ε_{B} 之间的关系. k=1 的曲线对应于图 6.1 的最近邻法错误率的上下界. 当 k 增大时, 上界逐渐逼近最优的贝叶斯决策下的平均错误率 ε_{B}.

由上述分析可知, 在 k 近邻法中, 我们希望采用较大的 k 值以减小错误率; 另一方面又要求 k 个近邻样本与待归类的样本 $\boldsymbol{x}$ 足够靠近, 以利于利用这些样本得到样本 $\boldsymbol{x}$ 的正确分类 (在式 (6.2.6) 的推导过程中利用了 $q(\omega_i|\boldsymbol{x})\cong q(\omega_i|\boldsymbol{x}')$ 的关系). 因此, 在实际使用 k 近邻法时, 一般要求满足 $k<<N$ 条件下取较大的 k 值.

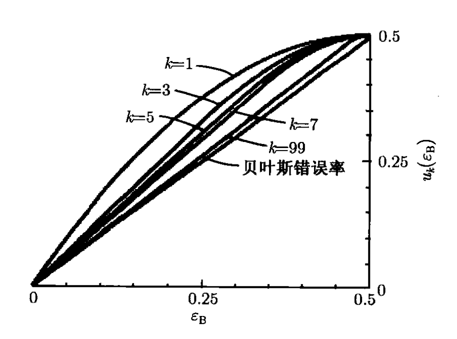


图 6.2　两类情形下 (c=2) k 近邻法错误率 ε_k 的上下界与贝叶斯决策错误率 ε_B 的关系

通常取 $k \approx \sqrt{N}$ 应该是不错的选择.

无论是近邻法还是 k-近邻法, 其基本思想和算法步骤都十分简单, 而且按照式 (6.1.5), 其错误率为

$$\varepsilon_B \leqslant \varepsilon \leqslant \varepsilon_B \left(2 - \frac{c}{c-1}\varepsilon_B\right).$$

考虑到一般情形下 ε_B 比较小, 可将括号中的第二项忽略, 近似地有

$$\varepsilon_B \leqslant \varepsilon \leqslant 2\varepsilon_B.$$

这就是常说的近邻法错误率介于 ε_B 和 $2\varepsilon_B$ 之间. 近邻法的这些优良性质使它成为模式分类的重要方法之一.

但是, 近邻法也存在以下不足:

(1) 需将训练样本集的所有 N 个样本存入计算机中, 每次决策都要计算待识别样本 $\boldsymbol{x}$ 与全部训练样本之间的距离并进行比较. 当 N 很大时, 存储量和计算量都很大.

(2) 所以以上分析的结果都是渐近的平均结果, 即要求 $N \to \infty$, 这在实际场合是无法实现的, 实际的结果与之存在差别.

(3) 虽然在所有情况下对未知样本都可以作出决策, 但当错误代价很大时, 会产生较大的风险.

6.3　剪辑近邻法

如所周知, 确定分类器错误率的方法之一是经验估计, 即利用所有样本的类别已知的训练集来估计错误分类的经验频数. 假如使用全部样本同时用来设计分类

器, 又用来估计错误率, 将由于样本集缺乏独立性使得错误率的估计偏于乐观. 如果将样本集分为两个独立的子集 —— 设计集和测试集, 用设计集设计分类器, 用测试集估计错误率, 这样得到的错误率应该是较为准确的. 上述的估计错误率的基本思想引出了剪辑近邻法.

6.3.1 两分剪辑近邻法

设训练样本集 N 个样本共分 c 个类别, 第 $i(i=1,2,\cdots,c)$ 类样本数为 N_i. 用集合

$$\boldsymbol{X}^N=\left\{\boldsymbol{X}_1^{N_1},\boldsymbol{X}_2^{N_2},\cdots,\boldsymbol{X}_c^{N_c}\right\}, \tag{6.3.1}$$

表示这 N 个样本, 其中每一类表示为

$$\boldsymbol{X}_i^{N_i}=\{\boldsymbol{x}_i^m\}, \qquad i=1,2,\cdots,c;m=1,2,\cdots,N_i. \tag{6.3.2}$$

剪辑近邻法的基本考虑是将决策过程分为两步. 第一步, 对训练样本集 N 个样本进行预分类, 剪辑掉被错分类的样本, 余下的样本构成剪辑样本集 $\boldsymbol{X}^{NE}$, 该过程称为剪辑；第二步利用剪辑样本集 $\boldsymbol{X}^{NE}$ 和近邻规则对未知样本 $\boldsymbol{x}$ 进行分类.

在两分剪辑近邻法中, 训练样本集 $\boldsymbol{X}^N$ 被分为两个独立的子集 —— 参考集(相当于错误率估计中的设计集)$\boldsymbol{X}^{NR}$ 和测试集 $\boldsymbol{X}^{NT}$. 两个子集中的样本不相重叠, 即 $N=NR+NT$. 参考集 $\boldsymbol{X}^{NR}$ 用以完成剪辑和设计任务, 而测试集 $\boldsymbol{X}^{NT}$ 则完成测试任务.

与近邻法相比, 剪辑近邻法增加了样本剪辑这一步骤, 所以, 需要讨论如何进行样本的剪辑.

1. *两类问题的最近邻法剪辑*

令参考集 $\boldsymbol{X}^{NR}$ 的样本用

$$\boldsymbol{X}^{NR}=\{\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_{NR}\} \tag{6.3.3}$$

表示. 测试集 $\boldsymbol{X}^{NT}$ 的样本用

$$\boldsymbol{X}^{NT}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_{NT}\}=\{\boldsymbol{x}_j\}, \qquad j=1,2,\cdots,NT \tag{6.3.4}$$

表示. 对于测试集 $\boldsymbol{X}^{NT}$ 的任一样本 $\boldsymbol{x}_j$, 其在参考集 $\boldsymbol{X}^{NR}$ 中的最近邻样本用 $\boldsymbol{y}'(\boldsymbol{x}_j)$ 表示. 所谓剪辑, 就是当测试集 $\boldsymbol{X}^{NT}$ 中的一个样本 $\boldsymbol{x}_j$ 与其在参考集 $\boldsymbol{X}^{NR}$ 中的最近邻样本 $\boldsymbol{y}'(\boldsymbol{x}_j)$ 不属于同一模式类时, 将它从 $\boldsymbol{X}^{NT}$ 中剪辑掉；当属于同一模式类时则予以保留. 对 $\boldsymbol{X}^{NT}$ 中所有样本完成剪辑步骤后, 形成剪辑样本集 $\boldsymbol{X}^{NTE}$.

然后, 对未知样本 $\boldsymbol{x}$ 的分类用剪辑样本集 $\boldsymbol{X}^{NTE}$ 和最近邻原则作分类决策.

可以证明, 剪辑最近邻法的渐近条件错误率与最近邻法的渐近条件错误率存在下述关系

$$\varepsilon_1^E(e|\boldsymbol{x}) = \frac{\varepsilon_1(e|\boldsymbol{x})}{2\left[1-\varepsilon_1(e|\boldsymbol{x})\right]}. \tag{6.3.5}$$

由上式可知, 剪辑最近邻法的错误率总是小于等于最近邻法的错误率, 即

$$\varepsilon_1^E(e) \leqslant \varepsilon_1(e). \tag{6.3.6}$$

特别当 $\varepsilon_1(e)$ 很小时, 例如 $\varepsilon_1(e) < 0.1$, 可推知

$$\varepsilon_1^E(e) \cong \varepsilon_1(e)/2. \tag{6.3.7}$$

由于最近邻法错误率 $\varepsilon_1(e)$ 的上界为 $2\varepsilon_{\mathrm{B}}$, 因此剪辑最近邻法的错误率接近贝叶斯错误率, 即

$$\varepsilon_1^E(e) \cong \varepsilon_{\mathrm{B}}. \tag{6.3.8}$$

2. 两类问题的 k-近邻法剪辑

上述最近邻剪辑法不难推广到 k 近邻的情况. 简单地说, 就是第一步用 k 近邻法进行剪辑, 第二步用剪辑样本集 $\boldsymbol{X}^{NTE}$ 和最近邻原则作分类决策. 用 k 近邻法进行剪辑就是, 对于测试集 $\boldsymbol{X}^{NT}$ 的任一样本 $\boldsymbol{x}_j$, 其在参考集 $\boldsymbol{X}^{NR}$ 中的 k 个近邻样本用 $\{\boldsymbol{y}_1'(\boldsymbol{x}_j), \boldsymbol{y}_2'(\boldsymbol{x}_j), \cdots, \boldsymbol{y}_k'(\boldsymbol{x}_j)\}$ 表示, 如果样本 $\boldsymbol{x}_j$ 与 $\{\boldsymbol{y}_1'(\boldsymbol{x}_j), \boldsymbol{y}_2'(\boldsymbol{x}_j), \cdots, \boldsymbol{y}_k'(\boldsymbol{x}_j)\}$ 样本中最多数的类别不一致, 则样本 $\boldsymbol{x}_j$ 从测试集 $\boldsymbol{X}^{NT}$ 中被剪辑掉.

可以证明此时有类似于式 (6.3.5) 所示的关系

$$\varepsilon_k^E(e|\boldsymbol{x}) = \frac{\varepsilon_1(e|\boldsymbol{x})}{2\left[1-\varepsilon_k(e|\boldsymbol{x})\right]}. \tag{6.3.9}$$

由于一般说来 k 近邻法的渐近条件错误率小于最近邻法的渐近条件错误率, 比较式 (6.3.9) 与式 (6.3.5) 可得

$$\varepsilon_k^E(e|\boldsymbol{x}) < \varepsilon_1^E(e|\boldsymbol{x}). \tag{6.3.10}$$

假定 $N \to \infty$ 时有 $k \to \infty$, 且 $k/N \to 0$ (即 k 足够大, 但 $k << N$), 则有

$$\lim_{k\to\infty} \varepsilon_k(e|\boldsymbol{x}) = \varepsilon_{\mathrm{B}}(e|\boldsymbol{x}).$$

利用

$$\varepsilon_1(e|\boldsymbol{x}) = 2\varepsilon_{\mathrm{B}}(e|\boldsymbol{x})\left[1-\varepsilon_{\mathrm{B}}(e|\boldsymbol{x})\right]$$

代入式 (6.3.9) 得

$$\lim_{k\to\infty} \varepsilon_k^E(e|\boldsymbol{x}) = \varepsilon_{\mathrm{B}}(e|\boldsymbol{x}).$$

两边取期望值得

$$\varepsilon_k^E(e) = \varepsilon_{\rm B}. \tag{6.3.11}$$

上式表明, k 近邻剪辑法当 $k \to \infty$ 时其错误率 $\varepsilon_k^E(e)$ 收敛于最优错误率 $\varepsilon_{\rm B}$. 这显然比最近邻剪辑法的特性为好.

3. 多类问题的近邻法剪辑

对于多类问题, 剪辑的效果将变得更好. 对于 c 类问题, 第一步用 k 近邻法进行剪辑, 第二步用剪辑样本集 $\boldsymbol{X}^{NTE}$ 和最近邻原则作分类决策. 若用 $P_k(\omega_l|\boldsymbol{x})$ 表示用 k 近邻法分配样本 $\boldsymbol{x}$ 为 ω_l 类的概率, 可以证明, 此时的错误率为

$$\varepsilon_{kc}^E(e|\boldsymbol{x}) = \varepsilon_k^E(e|\boldsymbol{x}) - \frac{\sum\limits_{i,j,l} P_k(\omega_i|\boldsymbol{x})P_k(\omega_j|\boldsymbol{x})P_k(\omega_l|\boldsymbol{x})}{1-\varepsilon_k(e|\boldsymbol{x})} \tag{6.3.12}$$

其中, $i=1,2,\cdots,c-1; j=i+1,\cdots,c; l=1,2,\cdots,c$, 并且 $l \neq i,j$.

当 c=2 时, 由于上式中 $P_k(\omega_l|\boldsymbol{x}) = 0$, 式 (6.3.12) 简化为式 (6.3.9) 的两类错误率. 在其他情况下, 式 (6.3.12) 右边第二项大于 0, 所以多类剪辑近邻法错误率 $\varepsilon_{kc}^E(e|\boldsymbol{x})$ 将小于两类剪辑近邻法错误率 $\varepsilon_k^E(e|\boldsymbol{x})$.

6.3.2 重复剪辑近邻法

如果训练样本集的样本数量足够多, 可以重复进行剪辑, 以提高近邻规则的分类性能. 可以证明, 对于两类问题, 利用剪辑最近邻法重复进行 m 次剪辑后再进行分类, 当 $m \to \infty$ 时,

$$\lim_{m\to\infty} \varepsilon_{m\times 1,1}(e|\boldsymbol{x}) = \varepsilon_{\rm B}(e|\boldsymbol{x}).$$

即当 m 充分大时, 其错误率渐近地收敛于最优错误率 $\varepsilon_{\rm B}$.

重复剪辑近邻法的一种实际算法被称为 MULTIEDIT 算法, 其计算步骤如下:

(1) 将样本集 $\boldsymbol{X}^N$ 随机地划分为 s 个子集 ($s \geqslant 3$), 即

$$\boldsymbol{X}^N = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_s\}.$$

(2) 对于所有的 $i(i=1,2,\cdots,s)$, 将 $\boldsymbol{X}_i$ 视为测试集, $\boldsymbol{X}_{(i+1){\rm Mod}(i)}$ 视为参考集, 其中

$$(i+1){\rm Mod}(s) = (i+1) - \left[\frac{i+1}{s}\right]s.$$

利用剪辑最近邻法对测试集 $\boldsymbol{X}_i$ 中的样本进行剪辑. 剪辑留下的样本, 构成剪辑样本集 $\boldsymbol{X}^{NE}$.

(3) 如果步骤 (2) 没有剪辑掉任何样本, 即 $\boldsymbol{X}^{NE}$ 与 $\boldsymbol{X}^N$ 相等, 则算法终止; 否则将 $\boldsymbol{X}^{NE}$ 视为"新"样本集 $\boldsymbol{X}^N$, 转向步骤 (1).

由于对样本集进行了随机划分，并在以后的每次迭代中，都是将前一步剪辑后的样本构成新的样本集，然后再对其重新随机划分，这就有效地避免了划分子集间的相互作用，从而保证了剪辑的独立性.

图 6.3~6.5 是利用 MULTIEDIT 算法划分两类样本的一个例子. 图中十字叉和圆圈表示两类的样本点，虚线表示贝叶斯最优决策面，实线为重复剪辑近邻法确定的边界，它是分段线性的. 图 6.3 是初始样本集，图 6.4 是一次剪辑后的剪辑样本集，图 6.5 是最终结果. 由图可知，剪辑过程是将两类边界附近的样本去除. 重复剪辑近邻法最终确定的边界与贝叶斯最优决策面十分接近.

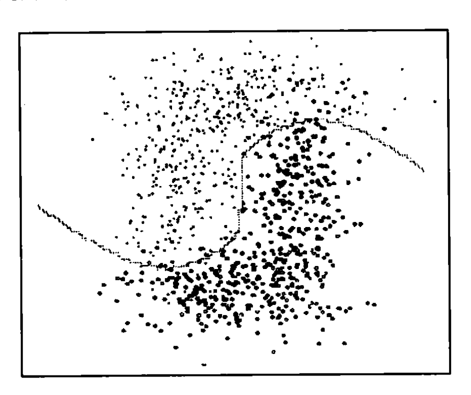


图 6.3　MULTIEDIT 算法划分两类样本：初始样本

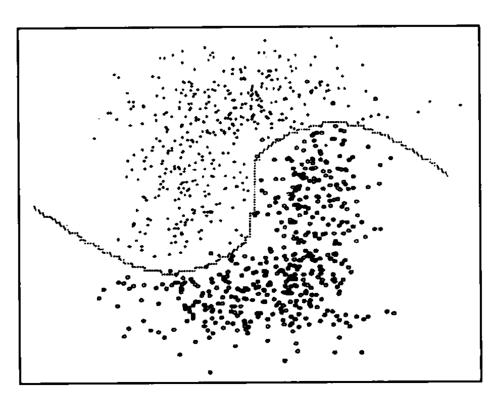


图 6.4　MULTIEDIT 算法划分两类样本：一次剪辑后的剪辑样本

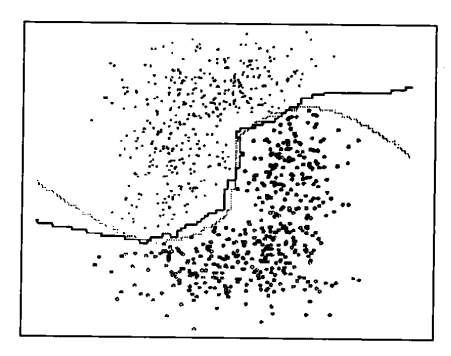


图 6.5 MULTIEDIT 算法划分两类样本：最终结果

6.4 可作拒绝决策的近邻法

利用近邻法进行分类时, 有时会出现决策风险很大的情况. 为了减小出现决策风险很大的情况的概率, 可引入拒绝决策的近邻法. 从 6.2 节我们知道, 在 k 近邻法中, 对于两类问题, 对于任意待归类的样本 $\boldsymbol{x}$, 它的 k 个近邻训练样本中有大于 $k_{\rm th}=0.5k\equiv tk$ 个样本属于某一类 $\omega_i(i=1,2)$, 则决策 $\boldsymbol{x}\in\omega_i$. 这种决策的拒绝率定义为 $1-t$. 可以想像, 如果将 t 值增大, 要求 $\boldsymbol{x}$ 的 k 个近邻训练样本中有更多的属于 ω_i 的样本时才决策 $\boldsymbol{x}\in\omega_i$, 那么该决策为错误的风险会减小. 这就是拒绝决策的近邻法的基本思想.

6.4.1 具有拒绝决策的 k 近邻法

对于两类问题, 具有拒绝决策的 k 近邻法可叙述如下. 给定阈值 t:

$$k_{\rm th}=tk,\qquad t>1/2,\tag{6.4.1}$$

对于任意待归类的样本 $\boldsymbol{x}$, 它的 k 个近邻样本中有大于等于 $k_{\rm th}$ 个样本属于某一类 $\omega_i(i=1,2)$, 则决策 $\boldsymbol{x}\in\omega_i$; 如果不满足以上条件, 则拒绝对样本 $\boldsymbol{x}$ 作归类决策.

$\boldsymbol{x}$ 的 k 个近邻样本中至少 $k_{\rm th}$ 个来自 ω_1 的渐近概率为

$$P(\omega_1|\boldsymbol{x})=\sum_{i=k_{\rm th}}^{k}C_k^i q(\omega_1|\boldsymbol{x})^i q(\omega_2|\boldsymbol{x})^{k-i}.\tag{6.4.2}$$

$P(\omega_1|\boldsymbol{x})$ 就是 $\boldsymbol{x}$ 决策为 ω_1 的概率, 其中 $q(\omega_i|\boldsymbol{x})$ 是给定 $\boldsymbol{x}$ 的后验概率. 当 $\boldsymbol{x}$ 的 k 个近邻样本中少于 $k_{\rm th}$ 个属于同一类别时拒绝对 $\boldsymbol{x}$ 作归类决策, 其概率为

$$P(\omega_0|\boldsymbol{x}) = \sum_{i=k-k_{\text{th}}+1}^{k_{\text{th}}-1} C_k^i q(\omega_1|\boldsymbol{x})^i q(\omega_2|\boldsymbol{x})^{k-i}. \tag{6.4.3}$$

$P(\omega_0|\boldsymbol{x})$ 就是 $\boldsymbol{x}$ 的渐近拒绝率, 它实际上建立了一个新的类别 —— 拒绝类 ω_0, 即 $\boldsymbol{x}$ 的 k 个近邻样本中有一部分样本既不归类为 ω_1, 也不归类为 ω_2.

由于决策必然为 $\omega_i (i = 0, 1, 2)$ 之一, 故有

$$P(\omega_0|\boldsymbol{x}) + P(\omega_1|\boldsymbol{x}) + P(\omega_2|\boldsymbol{x}) = 1 \tag{6.4.4}$$

决策错误率为

$$\varepsilon(e|\boldsymbol{x}) = q(\omega_1|\boldsymbol{x})P(\omega_2|\boldsymbol{x}) + q(\omega_2|\boldsymbol{x})P(\omega_1|\boldsymbol{x}) \tag{6.4.5}$$

决策拒绝率为

$$R(e|\boldsymbol{x}) = P(\omega_0|\boldsymbol{x}) = 1 - P(\omega_1|\boldsymbol{x}) - P(\omega_2|\boldsymbol{x}) \tag{6.4.6}$$

可以证明, 当 $k \to \infty$ 时, 上述渐近条件错误率和拒绝率分别收敛于具有拒绝阈值 $1 - t$ 的贝叶斯错误率和拒绝率.

6.4.2 具有拒绝决策的剪辑近邻法

具有拒绝决策的近邻法很容易推广到有剪辑的情况. 对于两类问题, 具有拒绝决策的剪辑近邻法可叙述如下.

给定 k 和式 (6.4.1) 的 k_{th} 值, 给定所有样本的类别已知的训练样本集 $\boldsymbol{X}^N$. 用以下步骤进行剪辑:

(1) 对于 $\boldsymbol{X}^N$ 中的每一个样本 $\boldsymbol{x}_i$, 从 $\boldsymbol{X}^N$ 中找出其 k 个近邻样本.

(2) 若 $\boldsymbol{x}_i$ 的 k 个近邻样本中至少有 k_{th} 个属于 ω_j 类, 则类别标志记为 $E_{\omega_j} = j (j = 1, 2)$ 类 ; 若不满足上述条件, 则类别标志记为 $E_{\omega_j} = 0$.

(3) 将 $E_{\omega_j} \neq 0$ 同时 $E_{\omega_j} \neq \theta_i$ 的样本从 $\boldsymbol{X}^N$ 中剪辑掉, 这里 θ_i 是样本 $\boldsymbol{x}_i$ 的已知类别标志, 也就是将步骤 (1), (2) 中错分的样本剪辑掉.

(4) 将 $E_{\omega_j} = 0$ 的样本归为拒绝类 ω_0. 这样原样本集 $\boldsymbol{X}^N$ 中的一部分被剪辑掉, 一部分建立了拒绝类 ω_0, 从而构成新的剪辑样本集 $\boldsymbol{X}^{NR}$.

(5) 利用剪辑样本集 $\boldsymbol{X}^{NR}$ 和最近邻法对未知样本 $\boldsymbol{x}$ 进行分类决策.

用 $\varepsilon^E(e)$ 和 $R^E(e)$ 分别表示剪辑方法的错误率和拒绝率, 它们与不考虑剪辑的错误率和拒绝率 $\varepsilon(e)$ 和 $R(e)$ 的关系如图 6.6 所示, 即剪辑后错误率减小而拒绝率增加.

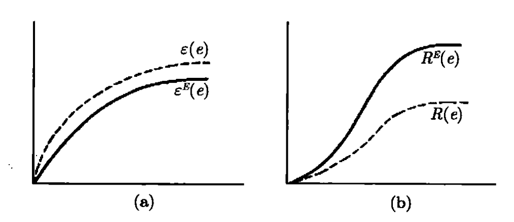


图 6.6 剪辑和非剪辑情况下的错误率和拒绝率

(a) 错误率; (b) 拒绝率

第七章　其他非线性判别方法

本章将讨论前面未曾涉及的、在实验数据分析中常见的一些非线性分类方法.

7.1　概率密度估计量方法

在本节的讨论中, 我们把问题局限于实验数据分析中的常见情形, 即观测数据仅分为信号和本底的两类问题.

7.1.1　基本思想

设样本的特征向量为 $\boldsymbol{x}=(x_1,\cdots,x_n)^{\mathrm{T}}$. 如果对于信号事例样本和本底事例样本, 其概率密度 $p_{\mathrm{S}}(\boldsymbol{x})$ 和 $p_{\mathrm{B}}(\boldsymbol{x})$ 均为已知, 则可以利用 $p_{\mathrm{S}}(\boldsymbol{x})$ 和 $p_{\mathrm{B}}(\boldsymbol{x})$ 构造判别量来判别未知样本的类别.

设未知样本 i 的特征向量为 $\boldsymbol{x}_i=(x_1(i),\cdots,x_n(i))^{\mathrm{T}}$, 概率密度估计量方法 (probability density estimator approach, PDE) 认为[43], 该样本属于信号类样本的概率可用下式表示:

$$y(\boldsymbol{x}_i)=\frac{p_{\mathrm{S}}(\boldsymbol{x}_i)}{p_{\mathrm{S}}(\boldsymbol{x}_i)+p_{\mathrm{B}}(\boldsymbol{x}_i)}. \tag{7.1.1}$$

该式的含义可作如下理解: $p_{\mathrm{S}}(\boldsymbol{x}_i)$ 表示特征向量取值 $\boldsymbol{x}_i$ 时样本属于信号事例的概率, $p_{\mathrm{B}}(\boldsymbol{x}_i)$ 表示特征向量取值 $\boldsymbol{x}_i$ 时样本属于本底事例的概率; 因此特征向量取值 $\boldsymbol{x}_i$ 的总概率为 $p_{\mathrm{S}}(\boldsymbol{x}_i)+p_{\mathrm{B}}(\boldsymbol{x}_i)$, 而 $y(\boldsymbol{x}_i)$ 表示特征向量取值 $\boldsymbol{x}_i$ 时样本属于信号事例的概率相对于总概率的比值. 这一比值也称为似然比估计量 (likelihood estimator), 可作为样本 i 的类别的判别函数. 对于信号样本 $\boldsymbol{x}$, 它被分类器判为信号事例的可能性应当大大高于被判为本底事例的可能性, 即 $p_{\mathrm{S}}(\boldsymbol{x})\gg p_{\mathrm{B}}(\boldsymbol{x})$, 故其 $y(\boldsymbol{x})\approx 1$. 反之, 对于本底样本 $\boldsymbol{x}$, 它被分类器判为信号事例的可能性应当大大低于被判为本底事例的可能性, 即 $p_{\mathrm{S}}(\boldsymbol{x})\ll p_{\mathrm{B}}(\boldsymbol{x})$, 故其 $y(\boldsymbol{x})\approx 0$. 因而, 设定一个阈值 $y(th)$, 决策规则可表示为

$$\begin{cases} y(\boldsymbol{x}_i)\geqslant y(th), & \text{样本 } i \text{ 判为信号;}\\ y(\boldsymbol{x}_i)< y(th), & \text{样本 } i \text{ 判为本底.}\end{cases} \tag{7.1.2}$$

这样, 当概率密度 $p_{\mathrm{S}}(\boldsymbol{x})$ 和 $p_{\mathrm{B}}(\boldsymbol{x})$ 均为已知时, 样本的分类问题就得到了解决.

实际上, 由式 (2.1.8) 知, 对于 $c=2$ 类问题, 基于最小错误率的贝叶斯决策规

则为

$$\begin{cases} \boldsymbol{x} \in \omega_1, & 当 \ \dfrac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)} > \dfrac{\pi(\omega_2)}{\pi(\omega_1)} \\ \boldsymbol{x} \in \omega_2, & 当 \ \dfrac{p(\boldsymbol{x}|\omega_1)}{p(\boldsymbol{x}|\omega_2)} < \dfrac{\pi(\omega_2)}{\pi(\omega_1)} \end{cases}$$

若将 $p(\boldsymbol{x}|\omega_1)$ 和 $p(\boldsymbol{x}|\omega_2)$ 分别表示为 $p_{\mathrm{S}}(\boldsymbol{x})$ 和 $p_{\mathrm{B}}(\boldsymbol{x})$, 并定义 $\alpha \equiv \pi(\omega_{\mathrm{B}})/\pi(\omega_{\mathrm{S}})$, 则上式可改写为

$$\begin{cases} \boldsymbol{x} \in \omega_{\mathrm{S}}, & 当 \ \dfrac{p_{\mathrm{S}}(\boldsymbol{x})}{p_{\mathrm{B}}(\boldsymbol{x})} > \alpha \\ \boldsymbol{x} \in \omega_{\mathrm{B}}, & 当 \ \dfrac{p_{\mathrm{S}}(\boldsymbol{x})}{p_{\mathrm{B}}(\boldsymbol{x})} < \alpha \end{cases} \tag{7.1.3}$$

经过简单的计算可知, 当取

$$y(th) = 1 - \frac{1}{1+\alpha} \tag{7.1.4}$$

时, 式 (7.1.1), 式 (7.1.2) 与式 (7.1.3) 等价. 可见 2 类问题基于最小错误率的贝叶斯决策式 (7.1.3) 是式 (7.1.1), 式 (7.1.2) 的特殊情形.

实际问题中, 概率密度 $p_{\mathrm{S}}(\boldsymbol{x})$ 和 $p_{\mathrm{B}}(\boldsymbol{x})$ 通常是未知的. 为此提出了直接用样本来估计总体分布的方法, 称之为估计分布的非参数法.

7.1.2 总体概率密度的非参数估计

我们的目的是利用训练样本集来估计样本空间任何一点的概率密度 $p(\boldsymbol{x})$, 这种估计用 $\hat{p}(\boldsymbol{x})$ 表示. 如果训练样本集来自某一类别 (如 ω_l 类, $l = 1, 2, \cdots, c$), 则估计结果为类条件概率密度 $\hat{p}(\boldsymbol{x}|\omega_l)$. 如果训练样本集来自 c 个类别, 又分不清哪个样本来自哪一类, 则估计结果 $\hat{p}(\boldsymbol{x})$ 为混合概率密度.

随机向量 $\boldsymbol{x}$ 落入区域 R 的概率 P 可表示为

$$P = \int_R p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{7.1.5}$$

若 $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N$ 是从概率密度 $p(\boldsymbol{x})$ 的总体分布中独立抽取的 N 个样本, 则有 k 个样本落入区域 R 的概率 P_k 服从二项分布, 即

$$P_k = C_N^k P^k (1 - P^k) \tag{7.1.6}$$

k 的期望值为 $NP, k/N$ 可以作为 P 的一个很好的估计, 也就是总体概率密度 $p(\boldsymbol{x})$ 在区域 R 上的好的估计. 为了求得总体概率密度 $p(\boldsymbol{x})$ 的估计 $\hat{p}(\boldsymbol{x})$, 设 $p(\boldsymbol{x})$ 连续, 并取区域 R 足够小, 以至于 $p(\boldsymbol{x})$ 在区域 R 的体积 V 内没有什么变化, 则有

$$P = \int_R p(x)\mathrm{d}\boldsymbol{x} \cong p(\boldsymbol{x})V \tag{7.1.7}$$

将 P 的估计值 k/N 代入, 得到任意 $\boldsymbol{x}$ 处概率密度 $p(\boldsymbol{x})$ 的估计 $\hat{p}(\boldsymbol{x})$ 为

$$\hat{p}(\boldsymbol{x})=\frac{k/N}{V}. \tag{7.1.8}$$

显然, $\hat{p}(\boldsymbol{x})$ 与总样本数 N、区域体积 V 和落入 V 的样本数 k 有关. 因为训练样本集的样本总数是有限的, 所以体积 V 不可能任意地小, 式 (7.1.7) 的近似使得式 (7.1.8) 的估计有一定的方差. 但理论上可以证明, 若满足以下三个条件, 则 $\hat{p}(\boldsymbol{x})$ 收敛于 $p(\boldsymbol{x})$:

$$\lim_{N\to\infty} V=0 \tag{7.1.9}$$

$$\lim_{N\to\infty} k=\infty \tag{7.1.10}$$

$$\lim_{N\to\infty} k/N=0 \tag{7.1.11}$$

在实际应用中, 式 (7.1.9) 要求体积 V 充分小, 但 V 中的样本数 k 按式 (7.1.10) 要求充分大, 同时, 按式 (7.1.11) 要求 k 又只占样本总数 N 的一小部分, 这样按式 (7.1.8) 确定的 $\hat{p}(\boldsymbol{x})$ 收敛于 $p(\boldsymbol{x})$, 即为 $p(\boldsymbol{x})$ 的好的近似.

1. *总体分布的 Parzen 核函数法估计*

Parzen 核函数法是一种常用的总体分布的非参数估计方法. 在 Parzen 核函数法中, 体积 V 以 N 的某个函数 (如 $V=1/\sqrt{N}$) 的关系不断缩小, 当 N 充分大时使 $\hat{p}(\boldsymbol{x})$ 收敛于 $p(\boldsymbol{x})$.

利用公式 (7.1.8), 并假定区域 R 是一个边长为 h 的超立方体, 即

$$V=h^n \tag{7.1.12}$$

定义核函数 (kernal function) $\varphi(\boldsymbol{u})(\boldsymbol{u}=(u_1,u_2,\cdots,u_n)^{\mathrm{T}})$

$$\varphi(\boldsymbol{u})=\begin{cases}1, & \text{当 } |u_j|\leqslant 1/2, j=1,2,\cdots,n\\ 0, & \text{其他}\end{cases} \tag{7.1.13}$$

利用式 (7.1.13) 可将落入超立方体 V 内的样本数 k 用解析式表示出来. 由于 $\varphi(\boldsymbol{u})$ 是以原点为中心的一个超立方体, 所以, 当样本 $\boldsymbol{x}_i$ 落在以 $\boldsymbol{x}$ 为中心、体积为 V 的超立方体内时, $\varphi(\boldsymbol{u})=\varphi\left(\dfrac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right)=1$; 而当样本 x_i 落在体积 V 之外时, $\varphi(\boldsymbol{u})=0$. 因此落入超立方体 V 内的样本数 k 为

$$k=\sum_{i=1}^{N}\varphi\left(\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right). \tag{7.1.14}$$

将它代入式 (7.1.8) 得

$$\hat{p}(\boldsymbol{x})=\frac{1}{N}\sum_{i=1}^{N}\frac{1}{V}\varphi\left(\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right). \tag{7.1.15}$$

上式即是 Parzen 核函数法估计总体分布的基本公式. 虽然这里该式是利用超立方体核函数推导出来的, 实际上也适用于其他核函数.

当然我们要问, 式 (7.1.15) 给定的估计量 $\hat{p}(\boldsymbol{x})$ 是不是一个合理的密度函数? 即它是否满足下述条件:

$$\int\hat{p}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}=1 \tag{7.1.16}$$

我们发现, 只要核函数满足以下条件:

$$\begin{cases}\varphi(\boldsymbol{u})\geqslant 0\\ \displaystyle\int\varphi(\boldsymbol{u})\mathrm{d}\boldsymbol{u}=1\end{cases} \tag{7.1.17}$$

即核函数本身具有密度函数的形式, 则 $\hat{p}(\boldsymbol{x})$ 一定是一个密度函数. 证明如下:

$$\begin{aligned}\int\hat{p}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}&=\int\frac{1}{N}\sum_{i=1}^{N}\frac{1}{V}\varphi\left(\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right)\mathrm{d}\boldsymbol{x}=\frac{1}{N}\sum_{i=1}^{N}\int\frac{1}{V}\varphi\left(\frac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right)\mathrm{d}\boldsymbol{x}\\&=\frac{1}{N}\sum_{i=1}^{N}\int\varphi(\boldsymbol{u})\mathrm{d}\boldsymbol{u}=\frac{1}{N}\cdot N=1.\end{aligned}$$

可见, 只要核函数 $\varphi(\boldsymbol{u})=\varphi\left(\dfrac{\boldsymbol{x}-\boldsymbol{x}_i}{h}\right)$ 具有密度函数的形式, 就可用式 (7.1.15) 给出总体分布密度函数 $p(\boldsymbol{x})$ 的估计量 $\hat{p}(\boldsymbol{x})$. 因此除了超立方体核函数外, 还可选择满足式 (7.1.17) 的其他核函数. 几个一维的例子如下:

$$\text{方窗核函数}\quad \varphi(u)=\begin{cases}1, & \text{当 } |u|\leqslant 1/2\\ 0, & \text{其他}\end{cases} \tag{7.1.18}$$

$$\text{正态核函数}\quad \varphi(u)=\frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{u^2}{2}\right\} \tag{7.1.19}$$

$$\text{指数核函数}\quad \varphi(u)=\exp\{-|u|\} \tag{7.1.20}$$

它们的图示见图 7.1.

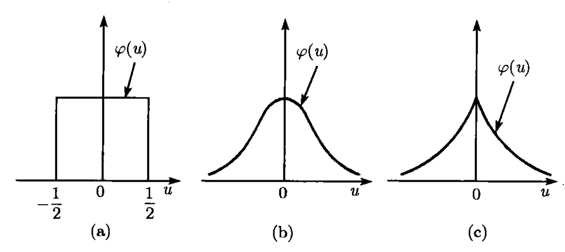


图 7.1　几种核函数

(a) 方窗核函数; (b) 正态核函数; (c) 指数核函数

在样本数 N 有限时, 窗宽 h 对估计量 $\hat{p}(\boldsymbol{x})$ 的品质有很大影响. 其原因分析如下. 定义函数 $\delta_V(\boldsymbol{x})$ 为

$$\delta_V(\boldsymbol{x}) = \frac{1}{V}\varphi\left(\frac{\boldsymbol{x}}{h}\right) \tag{7.1.21}$$

则 $\hat{p}(\boldsymbol{x})$ 可以视为 N 个 $\delta_V(\boldsymbol{x}-\boldsymbol{x}_i)$ 函数的平均值:

$$\hat{p}(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N}\delta_V(\boldsymbol{x}-\boldsymbol{x}_i) \tag{7.1.22}$$

当 h 很大, 即 $V=h^n$ 很大, $\delta_V(\boldsymbol{x}-\boldsymbol{x}_i)$ 的幅度就很小; 同时仅当 $|\boldsymbol{x}-\boldsymbol{x}_i| \gg h$ 时 $\delta_V(\boldsymbol{x}-\boldsymbol{x}_i)$ 与 $\delta_V(0)$ 差别才比较明显. 这时 $\hat{p}(\boldsymbol{x})$ 变成 N 个宽度很大且函数值变化缓慢的函数的叠加, 从而它是总体分布 $p(\boldsymbol{x})$ 的一个平均估计, 使估计的分辨能力降低. 反之, 当 h 很小, $\delta_V(\boldsymbol{x}-\boldsymbol{x}_i)$ 的幅度就很大, $\hat{p}(\boldsymbol{x})$ 变成 N 个以样本 $\boldsymbol{x}_i$ 为中心的尖峰函数的叠加, 使估计的统计涨落很大. 因此, 对于样本数 N 有限的实际情况, 窗宽 h 应当根据 N 的大小和总体分布 $p(\boldsymbol{x})$ 的形状来确定其适当的数值.

下面通过两个具体的例子来说明窗宽 h 和样本量 N 的大小如何影响估计量 $\hat{p}(\boldsymbol{x})$ 对于总体分布 $p(\boldsymbol{x})$ 的接近程度.

例一, 总体分布 $p(\boldsymbol{x})$ 为一维标准正态分布. 我们选择式 (7.1.19) 的正态核函数, 窗宽 h 选为 $h=h_0/\sqrt{N}$ 以考察样本量 N 对于估计量 $\hat{p}(\boldsymbol{x})$ 的作用, h_0 取三个值 0.25, 1, 4 以考察窗宽 h 对于估计量 $\hat{p}(\boldsymbol{x})$ 的作用. 所得的估计量 $\hat{p}(\boldsymbol{x})$ 如图 7.2 所示. 当 N=1 时, 所得的估计量 $\hat{p}(\boldsymbol{x})$ 与其说是总体分布 $p(\boldsymbol{x})$ 的估计, 不如说是核函数本身. 随着样本量 N 的增大, $\hat{p}(\boldsymbol{x})$ 逐渐逼近总体分布 $p(\boldsymbol{x})$, 但对于不同的 h_0 逼近的速度不同. 只有当样本量 N 趋于无穷, $\hat{p}(\boldsymbol{x})$ 才收敛于真实的总体分布 $p(\boldsymbol{x})$. 这说明要想得到较精确的估计, 必须要有大量的训练样本.

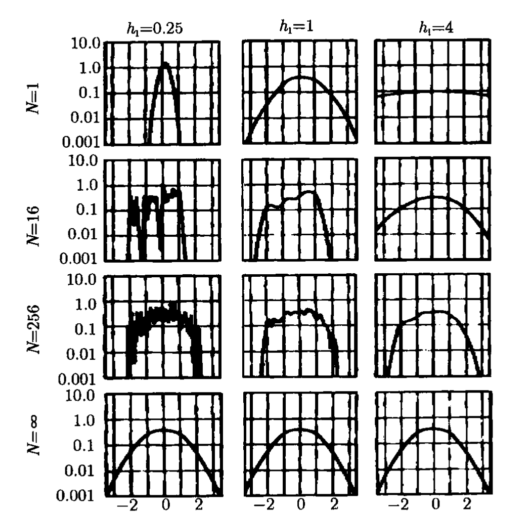


图 7.2 Parzen 核函数法估计一维正态分布

例二, 总体分布 $p(\boldsymbol{x})$ 为两个隔离的均匀分布构成的一维混合概率密度:

$$p(x)=\begin{cases}1, & -2.5<x<-2\\ 0.25, & 0<x<2\\ 0, & \text{其他}\end{cases}\tag{7.1.23}$$

正态核函数, 窗宽 h 的选择与例一相同. 所得的估计量 $\hat{p}(\boldsymbol{x})$ 如图 7.3 所示. 当 N=256 及 $h_0=1$ 时, $\hat{p}(\boldsymbol{x})$ 与总体分布 $p(\boldsymbol{x})$ 就较为接近了. 同样, 只有当样本量 N 趋于无穷, $\hat{p}(\boldsymbol{x})$ 才收敛于真实的总体分布 $p(\boldsymbol{x})$.

这两个例子反映了总体分布非参数估计方法的一些性质和存在的问题. 非参数估计的优点是它的普适性, 即对规则或不规则的分布, 单峰或多峰的分布都可以得到其密度函数的估计; 而且只要样本量充分大, 总可以收敛于任何复杂的未知密度函数. 其缺点是要想得到较为精确的估计, 需要远比参数估计方法多得多的样本量, 因此需要大量的计算时间和存储量.

最佳窗宽可以使渐近平均方差达到极小来求得[44], 对于高斯型核函数, 最佳窗宽为

$$h_G(j)=\left(\frac{4}{3}\right)^{1/5}\sigma(x_j)N^{-1/5},\qquad j=1,2,\cdots,n\tag{7.1.24}$$

其中 $\sigma(\boldsymbol{x}_j)$ 是 $\boldsymbol{x}$ 的第 j 个变量的标准差.

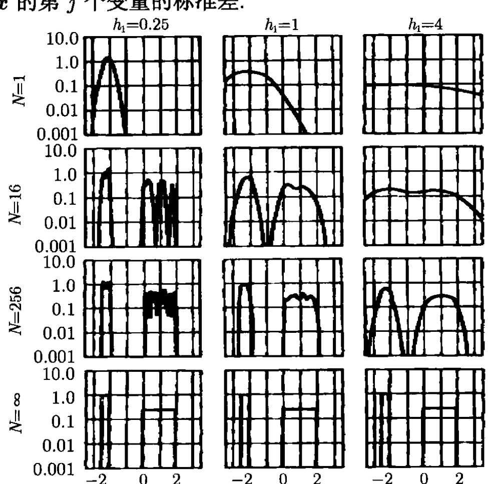


图 7.3　Parzen 核函数法估计两个隔离的一维均匀分布

也可以利用所谓的自适应方法来确定窗宽[44]. 在这种方法中窗宽不是一个固定常数, 而是随着总体分布 $p(\boldsymbol{x})$ 而变化. 设非自适应方法的窗宽为 h_{NA}, 则自适应方法的窗宽 h_A 为

$$h_A(j) = \frac{h_{NA}(j)}{\sqrt{p(x_j)}}, \qquad j = 1, 2, \cdots, n \tag{7.1.25}$$

在实际运算时, $p(\boldsymbol{x})$ 用其估计量 $\hat{p}(\boldsymbol{x})$ 作为近似.

2. 总体分布的 k_N 近邻估计

Parzen 核函数估计中存在的一个具体问题是, 对于有限的 N 值, $\hat{p}(\boldsymbol{x})$ 对于窗宽初值 h_0 的选择很敏感. h_0 过小, $\hat{p}(\boldsymbol{x})$ 的形状具有统计不稳定性; h_0 过大, $\hat{p}(\boldsymbol{x})$ 的形状偏于平坦, 不能反映总体分布 $p(\boldsymbol{x})$ 的细致结构. 为解决这一问题, 提出了总体分布的 k_N 近邻估计法.

Parzen 核函数法中, 体积 V 是样本数 N 的函数; 而 k_N 近邻估计法的基本思想是使体积 V 是样本点分布密度的函数, 而不是 N 的函数. 为了利用 N 个训练样本事例估计 $p(\boldsymbol{x})$, 先给定 N 的某个函数 k_N, 以 $\boldsymbol{x}$ 点为中心在其周围选择一个体积 V, 使 V 中的训练样本数为 k_N 个, 它们是样本 $\boldsymbol{x}$ 的 k_N 个近邻样本. 如果 $\boldsymbol{x}$ 点附近总体分布密度比较高, 则体积 V 比较小, 从而提高分辨能力; 如果 $\boldsymbol{x}$ 点附近密度比较低, 则体积 V 比较大.

k_N 近邻估计仍用基本估计式 (7.1.8), 即

$$\hat{p}(\boldsymbol{x}) = \frac{k_N/N}{V}.$$

假设条件仍然是式 (7.1.9)~(7.1.11). k_N 可取为 N 的某个函数, 例如 $k_N = k_0\sqrt{N}$, $k_0 \geqslant 1$ 为某个给定的整常数. 对于有限的 N 值, k_0 的选择也会影响到 $\hat{p}(\boldsymbol{x})$, 这一点与 Parzen 核函数估计中窗宽初值 h_0 的选择对 $\hat{p}(\boldsymbol{x})$ 的影响类似. 同样, 当样本量 N 趋于无穷, $\hat{p}(\boldsymbol{x})$ 收敛于真实的总体分布 $p(\boldsymbol{x})$. 图 7.4 显示了总体分布 $p(\boldsymbol{x})$ 为一维正态分布和两个隔离的一维均匀分布情形下 k_N 近邻估计的结果. k_N 近邻法也存在一般非参数估计的缺点, 即所需样本量很多. 测试表明, 对于一维总体分布, 用数百个样本一般可以得到较好的结果, 两维估计则需要数千个样本, 随着维数的增加, 样本数将急剧增多, 因而计算量和存储量很大.

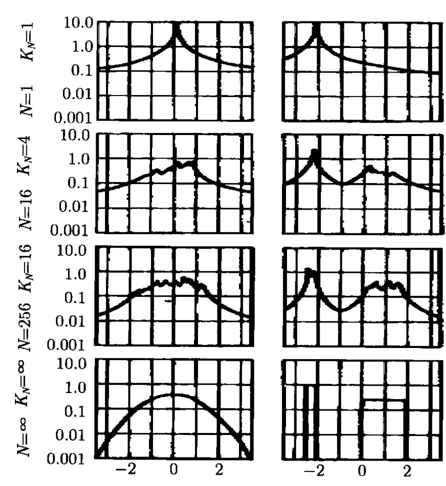


图 7.4 k_N 近邻法估计一维正态分布和两个隔离的一维均匀分布

7.1.3 投影似然比估计

所谓的投影似然比估计量 (projective likelihood estimator) 分类方法[45], 是指对于特征向量 $\boldsymbol{x} = (x_1, \cdots, x_n)^{\mathrm{T}}$ 的 n 个变量不相关联的情形, 这时概率密度 $p_{\mathrm{S}}(\boldsymbol{x})$ 和 $p_{\mathrm{B}}(\boldsymbol{x})$ 可以因子化为 n 个变量边沿概率密度的简单乘积, 即

$$p_{\mathrm{S(B)}}(\boldsymbol{x}) = p_{\mathrm{S(B)}}(x_1)p_{\mathrm{S(B)}}(x_2)\cdots p_{\mathrm{S(B)}}(x_n). \tag{7.1.26}$$

式中 $p_{\mathrm{S(B)}}(x_j), j=1,2,\cdots,n$ 为边沿概率密度, 它们是归一化的, 即有

$$\int_{-\infty}^{+\infty} p_{\mathrm{S(B)}}(x_j)\mathrm{d}x_j = 1, \qquad j=1,2,\cdots,n \tag{7.1.27}$$

在这种情形下, 多维随机变量的分析可以化作 n 个互不相关的一维随机变量的分析来处理.

设未知样本 i 的特征向量 $\boldsymbol{x}_i = (x_1(i),\cdots,x_n(i))^{\mathrm{T}}$ 第 j 个变量为 $x_j(i), j=1,2,\cdots,n$, 则样本 i 被视为信号事例的可能性由其似然值 $L_{\mathrm{S}}(i)$ 表征:

$$L_{\mathrm{S}}(i) = \prod_{j=1}^{n} p_{\mathrm{S},i}\left(x_j(i)\right) \tag{7.1.28}$$

样本 i 被视为本底事例的可能性由其似然值 $L_{\mathrm{B}}(i)$ 表征:

$$L_{\mathrm{B}}(i) = \prod_{j=1}^{n} p_{\mathrm{B},i}\left(x_j(i)\right) \tag{7.1.29}$$

样本 i 的似然比 $y_{\mathrm{L}}(i)$ 定义为

$$y_{\mathrm{L}}(i) = \frac{L_{\mathrm{S}}(i)}{L_{\mathrm{S}}(i)+L_{\mathrm{B}}(i)}. \tag{7.1.30}$$

这一似然比可作为样本 i 的类别的判别函数. 设定一个阈值 $y_{\mathrm{L}}(th)$, 决策规则可表示为

$$\begin{cases} y_{\mathrm{L}}(i) \geqslant y_{\mathrm{L}}(th), & \text{样本 } i \text{ 判为信号;} \\ y_{\mathrm{L}}(i) < y_{\mathrm{L}}(th), & \text{样本 } i \text{ 判为本底.} \end{cases} \tag{7.1.31}$$

这样, 当边沿概率密度 $p_{\mathrm{S(B)}}(x_j), j=1,2,\cdots,n$ 为已知时, 样本的分类问题就得到了解决. 当边沿概率密度 $p_{\mathrm{S(B)}}(x_j), j=1,2,\cdots,n$ 未知时, 可用 7.1.2 小节讨论的 Parzen 核函数法或 k_N 近邻法, 利用训练样本直接估计.

投影似然比估计方法的训练和应用思想简单、明确, 当边沿概率密度 $p_{\mathrm{S(B)}}(x_j)$, $j=1,2,\cdots,n$ 为已知时, 其计算速度很快, 适用于大数据样本的分类问题. 当然, 对于绝大多数实际问题, 其边沿概率密度是未知的, 需要用训练样本来确定. 为了得到概率密度的好的近似, 训练样本量需要很大, 因而计算量和存储量很大. 但这样的计算只需进行一次. 一旦边沿概率密度得以确定, 当应用于实际数据的分类时, 计算便十分简单. 投影似然比方法的主要缺陷是它没有考虑特征向量 $\boldsymbol{x}$ 各变量 $x_j, j=1,2,\cdots,n$ 之间的相互关联, 而这种关联在绝大多数实际问题中总是存在的. 这就使得投影似然比分类方法的错分率总是比较大, 并且具有某种不可控制性. 这一缺陷极大地限制了它的实际应用.

7.1.4 多维概率密度估计

对于投影似然比分类方法的改进自然会想到利用信号样本和本底样本的多维概率密度 $p_{\mathrm{S}}(\boldsymbol{x})$ 和 $p_{\mathrm{B}}(\boldsymbol{x})$ 来判别样本的类别. 这样就能穷尽特征向量 $\boldsymbol{x} = (x_1, \cdots, x_n)^{\mathrm{T}}$ 的全部信息, 因此分类器性能可达到最优. 为此需要有数量无穷大的信号样本集和本底样本集. 这实际上是无法实现的.

T. Carli 和 B. Koblits[43] 提出了一种简单的利用训练样本估计多维 $p(\boldsymbol{x})$ 的方法, 称为 PDE-RS (PDE range search) 方法, 它的基本思想即是前面叙述的总体分布的 k_N 近邻估计.

设信号/本底训练样本集的总样本数分别为 N_{S} 和 N_{B}. 又设待分类样本为 $\boldsymbol{x}_i$, 其近邻体积为 V, V 中的信号/本底事例数分别为 $n_{\mathrm{S}}(i,V)$ 和 $n_{\mathrm{B}}(i,V)$. 应用 k_N 近邻估计公式即 $\hat{p}_{\mathrm{S(B)}}(\boldsymbol{x}_i) = n_{\mathrm{S(B)}}(i,V)/N_{\mathrm{S(B)}}V$(参见式 7.1.8) 并代入式 (7.1.1), 立即得到

$$y(\boldsymbol{x}_i, V) = \frac{n_{\mathrm{S}}(i,V)}{n_{\mathrm{S}}(i,V) + a n_{\mathrm{B}}(i,V)}, \quad a \equiv \frac{N_{\mathrm{S}}}{N_{\mathrm{B}}} \tag{7.1.32}$$

似然比 $y(\boldsymbol{x}_i, V)$ 是待分类样本属于信号事例的概率密度在 $\boldsymbol{x}_i$ 附近的局部估计, 可作为样本 $\boldsymbol{x}_i$ 类别的判别函数. 设定一个阈值 $y(th)$, 决策规则可表示为

$$\begin{cases} y(\boldsymbol{x}_i, V) \geqslant y(th), & \text{样本 } \boldsymbol{x}_i \text{ 判为信号;} \\ y(\boldsymbol{x}_i, V) < y(th), & \text{样本 } \boldsymbol{x}_i \text{ 判为本底.} \end{cases} \tag{7.1.33}$$

对于信号训练样本, $y(\boldsymbol{x}_i, V)$ 在 1 附近将出现峰值; 对于本底训练样本, $y(\boldsymbol{x}_i, V)$ 在 0 附近将出现峰值. 这种估计中忽略了近邻体积 V 中的概率密度的变化, 因而是一种平均估计. PDE-RS 方法实际上是 k 近邻法的一个变种.

由式 (7.1.32), 容易求得 $y(\boldsymbol{x}_i, V)$ 的统计不确定性:

$$\sigma_y = \left[\left(\frac{a n_{\mathrm{B}}(i,V)}{[n_{\mathrm{S}}(i,V) + a n_{\mathrm{B}}(i,V)]^2}\sigma_{n_{\mathrm{S}}}\right)^2 + \left(\frac{a n_{\mathrm{S}}(i,V)}{[n_{\mathrm{S}}(i,V) + a n_{\mathrm{B}}(i,V)]^2}\sigma_{n_{\mathrm{B}}}\right)^2\right]^{1/2} \tag{7.1.34}$$

式中 $\sigma_{n_{\mathrm{S}}}$ 和 $\sigma_{n_{\mathrm{B}}}$ 是 V 中的信号/本底事例数 $n_{\mathrm{S}}(i,V)$ 和 $n_{\mathrm{B}}(i,V)$ 的统计不确定性.

7.1.5 近邻体积中样本数的确定

在概率密度估计法中, 当利用训练样本和式 (7.1.8) 估计 $\boldsymbol{x}$ 处的概率密度 $\hat{p}_{\mathrm{S}}(\boldsymbol{x})$ 和 $\hat{p}_{\mathrm{B}}(\boldsymbol{x})$ 时, 以及 PDE-RS 法利用式 (7.1.32) 计算待分类样本 $\boldsymbol{x}$ 的似然比 $y(\boldsymbol{x}, V)$ 时, 都要计算近邻体积 V 内的 (信号/本底) 训练样本数. 显然, 这一计算需要对不同的 $\boldsymbol{x}$ 值多次进行, 因此需要研发一种适合于计算机的算法, 能够高效地计算近邻体积 V 内的 (信号/本底) 训练样本数.

有两种算法可以完成这种计算. 第一种算法基本思想极为简单, 即将 $\boldsymbol{x}$ 的整个空间分成若干个子区间, 记下每个子区间内的样本数, 将所有子区间的位置和样本数信息以列表的方式存入计算机内存. 当要计算样本 $\boldsymbol{x}$ 的近邻体积 V 内的样本数时, 只要对属于 V 内的所有子区间内的样本数求和即可. 显然为了达到足够好的精度, 子区间体积应当比近邻体积 V 明显地小. 当训练样本数 N 很大, 特征向量 $\boldsymbol{x}$ 维数很高时, 所需的内存量很大; 并且需要知道训练样本集的 $\boldsymbol{x}$ 的 n 个变量的上下界.

另一种算法称为二叉树搜索算法 (binary tree search algorithm, BTSA)[46], 这是一种常用的更为有效的算法, 它不需要知道训练样本集的 $\boldsymbol{x}$ 的 n 个变量的上、下界. 在 BTSA 中, 对 N 个信号训练样本和 N 个本底训练样本分别建立二叉树 T_{S} 和 T_{B} 存储它们的信息.

我们用图 7.5 来说明二维特征向量样本集的二叉树的构建[43]. 假定共有 N=7 个信号样本用来构建二叉树 T_{S}. 其中的数字表示样本编号, 它们是随机地指定的. 样本的位置如图 7.5(a) 所示. 样本 1 中的 $\boldsymbol{e}_1(x_1,x_2)$ 被指定为 T_{S} 的第一层节点. 样本 2 中的 $\boldsymbol{e}_2(x_1,x_2)$ 为第二层节点, 在决定样本 2 为样本 1 的 "右" 节点还是 "左" 节点时, 需要比较 $\boldsymbol{e}_1(x_1,x_2)$ 和 $\boldsymbol{e}_2(x_1,x_2)$ 中 x_1 坐标值的大小, $\boldsymbol{e}_2$ 的 x_1 大, 样本 2 被指定为右节点. 样本 3 的 x_1 大于样本 1 的 x_1, 样本 3 应被指定为样本 1 的右节点, 但样本 2 已经占据了第二层节点的位置, 所以样本 3 被指定为第三层节点, 而且因为样本 3 的 x_2 大于样本 2 的 x_2, 故样本 3 被指定为样本 2 的右节点. 哪一个变量 (x_1 或 x_2) 用来比较以决定 "右" 节点还是 "左" 节点取决于节点的层数, 第一层比较 x_1, 第二层比较 x_2, 第三层比较 x_1(因为特征向量只有两个分量), 第四层比较 x_2, …… 以此类推, 可将所有 7 个样本构成图 7.5(b) 所示的 (信号) 二叉树. 对于高维的特征向量样本集, 可用类似的方法构建二叉树. 对于 N 个样本的训练集, 构建二叉树所需的时间 $t \propto N\log_2(N)$.

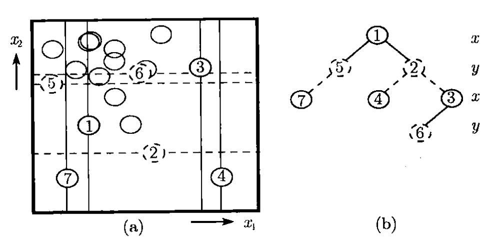


图 7.5 二叉树搜索算法示意图 ($\boldsymbol{x}$ 为二维特征向量)

(a) 数字 1~7 的圆圈标记信号样本; (b) 信号二叉树 T_{S} 的构建

当利用这样的二叉树 T 确定任意样本 $\boldsymbol{x}$ 的近邻体积 V 内的样本数 k 时, 通

过比较 V 的边界与 T 内节点的坐标来决定哪些训练样本应当被包含在 V 内, 再通过简单的计数即求得 k 值[43]. 所需的时间仅为 $t \propto N\log_2(N)$.

7.1.6 概率密度估计法与神经网络的性能对比

我们通过三个具体的例子来对比概率密度估计的 PDE-RS 方法与人工神经网络的性能. 应当强调指出, 这种性能对比仅针对这里的具体问题, 它们能否代表 PDE-RS 方法与 ANN 方法的一般性能有待研究.

1. **例一, 两维特征向量 $\boldsymbol{x}=(x_1,x_2)^{\mathrm{T}}$,$x_1$ 与 x_2 不相关**

信号样本为二维正态分布, 均值和标准差为

$$\bar{x}_{1\mathrm{S}}=4,\quad \bar{x}_{2\mathrm{S}}=3.5,\quad \sigma_{1\mathrm{S}}=\sigma_{2\mathrm{S}}=0.75.$$

本底样本为二维正态分布, 均值和标准差为

$$\bar{x}_{1\mathrm{B}}=3,\quad \bar{x}_{2\mathrm{B}}=4.5,\quad \sigma_{1\mathrm{B}}=\sigma_{2\mathrm{B}}=1.$$

图 7.6(a) 显示了信号样本和本底样本在 (x_1,x_2) 平面上的分布.

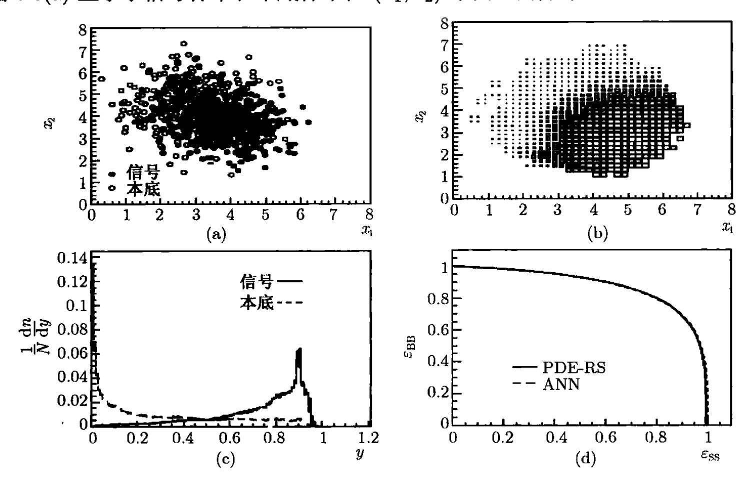


图 7.6 PDE-RS 法与神经网络 (ANN) 的性能对比

(a) 信号样本和本底样本在 (x_1,x_2) 平面上的分布; (b) PDE-RS 法用训练样本估计的 $\hat{p}(\boldsymbol{x})$; (c) PDE-RS 法得到的似然比 $y(\boldsymbol{x})$ 分布; (d) PDE-RS 法和 ANN 的 $\varepsilon_{\mathrm{SS}}$-$\varepsilon_{\mathrm{BB}}$ 关系曲线对比

我们以 N=100,000 个信号事例训练样本和 100,000 个本底事例训练样本用 PDE-RS 方法估计总体的概率密度, 近邻体积取为 $V=0.18\times0.18$, 得到的估计 $\hat{p}(\boldsymbol{x})$ 如图 7.6(b) 所示. 信号样本和本底样本的似然比 $y(\boldsymbol{x})$ 则如图 7.6(c) 所示. 图

7.6(d) 的横坐标 ε_{SS} 为一个信号事例被分类器判为信号事例的概率, 即信号事例的判选效率; 纵坐标为 $\varepsilon_{BB}=1-\varepsilon_{SB}$, 称为本底排除率, 其中 ε_{BB} 为一个本底事例被分类器正确地判为本底事例的概率, ε_{SB} 为一个本底事例被分类器错判为 "信号" 事例的概率. 如果一个分类器对于所有的信号/本底事例都能正确地分类, 则 $\varepsilon_{SS}=1$, $\varepsilon_{BB}=1$. 这时, ε_{SS}-ε_{BB} 关系曲线下的面积 A 为边长 1 的正方形, 即 A=1. 实际结果是 $A_{\text{PDERS}}=0.876\pm0.01$. 误差由式 (7.1.34) 的 σ_y 推算得到.

如果用三层前向 BP 网络 (参见 5.3 节) 来求解同样的问题, 其中隐含层的节点数取为 10, 则有 $A_{\text{ANN}}=0.877$. 由图 7.6(d) 知道, PDE-RS 法和 ANN 的 ε_{SS}-ε_{BB} 关系曲线基本上完全重合, 可见在本例中, PDE-RS 法和 ANN 的性能是相似的.

2. **例二, 两维特征向量 $\boldsymbol{x}=(x_1,x_2)^{\mathrm{T}}$,$x_1$ 与 x_2 强烈关联**

信号样本为半径 $r=G(3,0.5)$ 的正态 (高斯) 分布, 均值和标准差为 3 和 0.5. 本底样本为半径 $r=G(3,0.75)$ 的正态分布.

图 7.7(a) 显示了信号样本和本底样本在 (x_1,x_2) 平面上的分布.

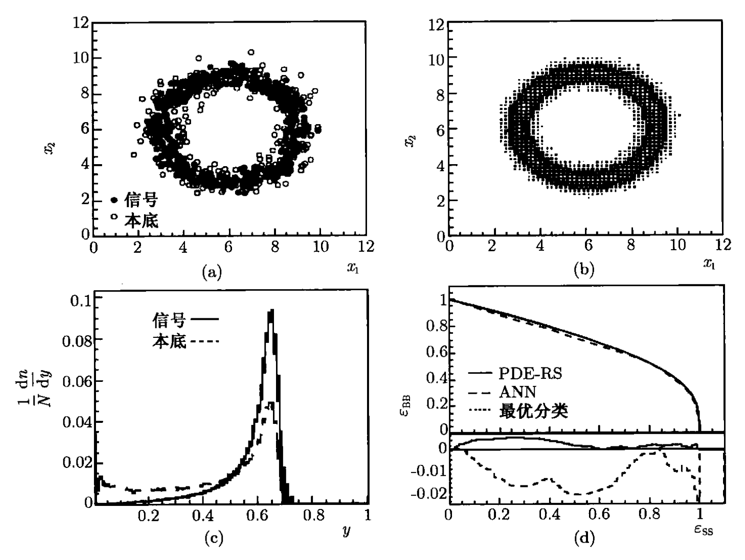


图 7.7　PDE-RS 法与神经网络 (ANN) 的性能对比

(a) 信号样本和本底样本在 (x_1,x_2) 平面上的分布; (b) PDE-RS 法用训练样本估计的 $\hat{p}(\boldsymbol{x})$; (c) PDE-RS 法得到的似然比 $y(\boldsymbol{x})$ 分布; (d) PDE-RS 法和 ANN 的 ε_{SS}-ε_{BB} 关系曲线对比 图下部的实线表示 PDE-RS 法与最优解的 ε_{SS}-ε_{BB} 曲线的差别, 虚线表示 ANN 法与最优解的 ε_{SS}-ε_{BB} 曲线的差别

我们以 N=100,000 个信号事例训练样本和 100,000 个本底事例训练样本用 PDE-RS 方法估计总体的概率密度, 近邻体积取为 $V = 0.12 \times 0.12$, 得到的估计 $\hat{p}(\boldsymbol{x})$ 如图 7.7(b) 所示. 信号样本和本底样本的似然比 $y(\boldsymbol{x})$ 则如图 7.7(c) 所示. 图 7.7(d) 为 PDE-RS 法和 ANN 的 ε_{SS}-ε_{BB} 关系曲线的对比. $A_{\text{PDERS}} = 0.708 \pm 0.031$. 用三层前向 BP 网络求解同样的问题, 隐含层的节点数取为 10, 则有 $A_{\text{ANN}} = 0.691$.

如果将两维特征向量 $\boldsymbol{x} = (x_1, x_2)^{\text{T}}$ 转化到极坐标中, 则成为一个一维 (半径 r) 样本的分类问题, 因此可以求得分类问题的最优解. 图 7.7(d) 的下部给出了 PDE-RS 法 (ANN 法) 与最优解的 ε_{SS}-ε_{BB} 曲线的差别. PDE-RS 法与最优解的差别比 ANN 与最优解的差别要小, 即 PDE-RS 法与最优解更接近.

利用同一台计算机, PDE-RS 法完成上述计算仅需 224s, 而完成同样的工作需要 34.6h 才能构建一个 ANN, 而要求得 ANN 的权值则需要多次构建 ANN.

本例说明, 对于特征向量各变量高度关联的数据, 无论在信号/本底鉴别性能上, 还是在计算时间上, PDE-RS 法均比 ANN 法优越.

3. 例三, 5 维特征向量 $\boldsymbol{x} = (x_1, x_2, \cdots, x_5)^{\text{T}}$, 各变量有中等的关联

信号样本的特征向量 $\boldsymbol{x}_{\text{S}}$ 为 $\boldsymbol{x}_{\text{S}} = M\boldsymbol{x}'_{\text{S}}$, 其中

$$M = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}, \qquad \boldsymbol{x}'_{\text{S}} = \begin{pmatrix} G(4,1) \\ G(1,1) \\ G(2,1.5) \\ G(2,1) \\ G(1.5,2) \end{pmatrix}$$

$G(m, \sigma)$ 表示均值 m, 标准差 σ 的正态随机变量. 本底样本的特征向量 $\boldsymbol{x}_{\text{B}}$ 为 $\boldsymbol{x}_{\text{B}} = M\boldsymbol{x}'_{\text{B}}$, 其中 $\boldsymbol{x}'_{\text{B}} = (G(4,1), G(2,1), G(3,1.5), G(1,1), G(0.5,1))^{\text{T}}$.

图 7.8(a) 显示了信号样本和本底样本 5 维特征向量在 (x_1, x_2) 平面上的投影分布. 以 N=500,000 个信号事例训练样本和 500,000 个本底事例训练样本用 PDE-RS 方法估计总体的概率密度, 近邻体积取为 $V = 1.2^5$ 超立方体, 信号样本和本底样本的似然比 $y(\boldsymbol{x})$ 如图 7.8(b) 所示. 图 7.8(c) 为 PDE-RS 法和 ANN 的 ε_{SS}-ε_{BB} 关系曲线的对比. $A_{\text{PDERS}} = 0.906 \pm 0.008$. 用三层前向 BP 网络求解同样的问题, 隐含层的节点数取为 10, 以 N=500,000 个信号事例训练样本和本底事例训练样本对 ANN 训练 1000 次, 得到 $A_{\text{ANN}} = 0.910$.

图 7.9(a) 显示了 PDE-RS 法在信号效率 $\varepsilon_{SS} = 0.7$ 情形下分辨能力 r 与近邻体积 V 的边长 h 的关系曲线. 分辨能力 r 定义为信号效率与本底效率之比

$$r = \varepsilon_{SS}/\varepsilon_{SB}$$

当 h 过大, 利用样本估计得到的 $\hat{p}(\boldsymbol{x})$ 不能反映真实分布 $p(\boldsymbol{x})$ 的细致结构, 导致分辨能力下降.h 过小, 近邻体积 V 内样本数过少, 统计涨落过大导致分辨能力下降. 两者之间是一个 r 值大体不变的平台区. 测试表明, 在一个适当的窗宽 (边长) 范围内, 分辨能力不随窗宽而变化是 PDE-RS 方法的一般性质. 这一范围随着训练样本数的增加而增大, 因此, 利用大训练样本能够改善 PDE-RS 分类器的信号/本底分辨性能. 图 7.9(b) 显示了 PDE-RS 法的计算时间与窗宽 h 的关系. 当 h 增大时计算时间明显增大. 另一方面, 如果加大训练样本数 N, 计算时间只随 N 对数地增大, 即 $t \sim \log_2(N)$. 因此可利用增加训练样本数 N 并减小 h 来达到同样的分辨能力. 例如图 7.9(b) 中 N=100k, h=2.5 与 N=500k, h=0.8 (图 (b) 中用箭头相连的两个圆圈标记) 有相同的分辨能力, 但后者的计算时间仅为前者的 1/10. 因此, 利用大训练样本和较小的窗宽能明显减小 PDE-RS 分类器的计算时间.

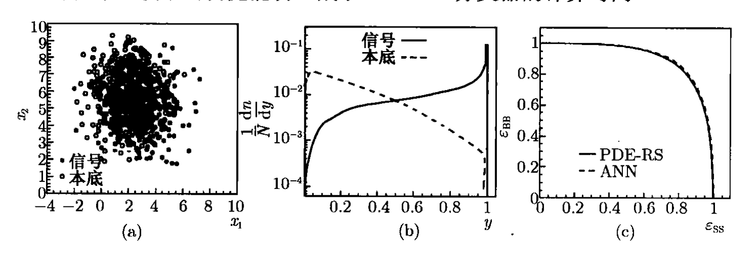


图 7.8　PDE-RS 法与神经网络 (ANN) 的性能对比

(a) 信号样本和本底样本在 (x_1, x_2) 平面上的投影分布; (b) PDE-RS 法得到的似然比 $y(\boldsymbol{x})$ 分布; (c) PDE-RS 法和 ANN 的 ε_{SS}-ε_{BB} 关系曲线对比

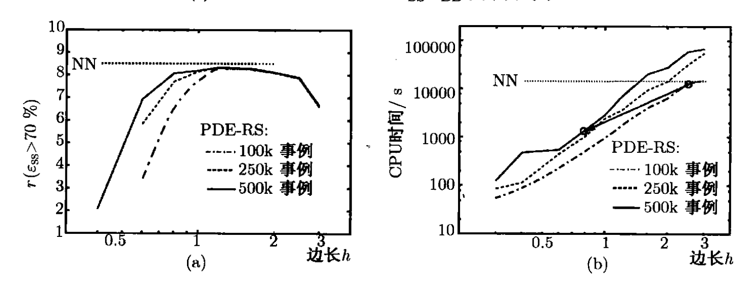


图 7.9　PDE-RS 法与神经网络 (ANN) 的性能对比

(a) PDE-RS 法和 ANN 在 $\varepsilon_{SS} = 0.7$ 情形下分辨能力 r 与近邻体积边长 h 的关系;
(b) PDE-RS 法和 ANN 的计算时间与 h 的关系

ANN 分类器的性能和计算时间与窗宽 h 无关, 因为它不是 ANN 分类器的参数. 图 7.9 中 ANN 的直线也是利用三层前向 BP 网络得到的, 其隐含层的节点数

为 10. 如图 7.9(b) 所示, 如果采用 N=500k, h=0.8 的 PDE-RS 分类器, 它的计算时间仅为 ANN 的 1/10, 而两者的性能是相近的.

7.2 $\boldsymbol{H}$ 矩阵判别

$\boldsymbol{H}$ 矩阵判别方法[45] 的起源可追溯到 Fisher[17] 和 Mahalanobis[47] 对于两个多维正态总体判别的工作.

设有由 $N_{\rm S}$ 个信号样本和 $N_{\rm B}$ 个本底样本构成的训练样本集, 它们服从 n 维正态分布, 即样本为 n 维特征向量. 它们的均值可由样本平均估计:

$$\bar{x}_{U,j} = \frac{1}{N_U}\sum_{i=1}^{N_U} x_j(i), \qquad j = 1, 2, \cdots, n \tag{7.2.1}$$

式中, U 为 S 或 B 分别对应于信号和本底, $x_j(i)$ 表示第 i 个样本第 j 个变量的值. 信号和本底总体的协方差矩阵可由样本协方差矩阵估计:

$$V_{U,lm} = \frac{1}{N_U - 1}\sum_{k=1}^{N_U} (x_l(k) - \bar{x}_{U,l})\,(x_m(k) - \bar{x}_{U,m}) \tag{7.2.2}$$

协方差矩阵 $\boldsymbol{V}_U$ 的逆矩阵被称为 $\boldsymbol{H}$ 矩阵, 即

$$\boldsymbol{H}_U = \boldsymbol{V}_U^{-1} \tag{7.2.3}$$

对于任意待判别类别的样本 $\boldsymbol{x}_i$, 构造信号和本底的 χ^2 估计量:

$$\chi_U^2(i) = \sum_{l,m=1}^{n} (x_l(i) - \bar{x}_{U,l})\, H_{U,lm}\, (x_m(i) - \bar{x}_{U,m}) \tag{7.2.4}$$

由式 (1.3.14) 知 $\chi_U^2(i)$ 即是样本 $\boldsymbol{x}_i$ 到 $N_{\rm S}$ 个信号样本和 $N_{\rm B}$ 个本底样本均值的 Mahalanobis (马氏) 距离. 马氏距离考虑了样本的特征向量分量的统计特性, 特别是考虑了各分量的相关性影响. 它是 $\boldsymbol{x}_i$ 到信号 (本底) 样本集间平均距离远近的度量, 即 $\chi_U^2(i)$ 越小, $\boldsymbol{x}_i$ 到 (信号/本底) 样本集间平均距离越近. 这样, 就可以根据 $\chi_{\rm S}^2(i)$ 和 $\chi_{\rm B}^2(i)$ 值来决定样本 $\boldsymbol{x}_i$ 的类别. 直观地, 可以预期, 判别规则可以是

$$\begin{cases} \chi_{\rm S}^2(i) < \chi_{\rm B}^2(i), & \boldsymbol{x}_i \text{ 判为信号;} \\ \chi_{\rm S}^2(i) > \chi_{\rm B}^2(i), & \boldsymbol{x}_i \text{ 判为本底.} \end{cases} \tag{7.2.5}$$

利用 $\chi_{\rm S}^2(i)$ 和 $\chi_{\rm B}^2(i)$ 来构造 $\boldsymbol{H}$ 矩阵判别方法对于样本 $\boldsymbol{x}_i$ 的判别函数 $g_{\rm H}(i)$:

$$g_{\rm H}(i) = \frac{\chi_{\rm B}^2(i) - \chi_{\rm S}^2(i)}{\chi_{\rm B}^2(i) + \chi_{\rm S}^2(i)} \tag{7.2.6}$$

决策面方程为

$$g_{\mathrm{H}}(i)=\frac{\chi_{\mathrm{B}}^2(i)-\chi_{\mathrm{S}}^2(i)}{\chi_{\mathrm{B}}^2(i)+\chi_{\mathrm{S}}^2(i)}=0 \tag{7.2.7}$$

决策规则为

$$\begin{cases} g_{\mathrm{H}}(i)>0, & \boldsymbol{x}_i \text{ 判为信号;} \\ g_{\mathrm{H}}(i)<0, & \boldsymbol{x}_i \text{ 判为本底.} \end{cases} \tag{7.2.8}$$

显然, 式 (7.2.8) 与式 (7.2.5) 是等价的, 判别函数 $g_{\mathrm{H}}(i)$ 中分母 $\chi_{\mathrm{B}}^2(i)+\chi_{\mathrm{S}}^2(i)$ 只起到归一化常数的作用, 即使得

$$g_{\mathrm{H}}(i)\in[-1,+1]$$

因为当 $\boldsymbol{x}_i=\bar{\boldsymbol{x}}_{\mathrm{S}}$ ($\bar{\boldsymbol{x}}_{\mathrm{S}}$ 为信号样本集均值向量) 时, $\chi_{\mathrm{S}}^2(i)=0$, $g_{\mathrm{H}}(i)$ 达到极大且等于 1. 反之, $\boldsymbol{x}_i=\bar{\boldsymbol{x}}_{\mathrm{B}}$ ($\bar{\boldsymbol{x}}_{\mathrm{B}}$ 为本底样本集均值向量) 时, $\chi_{\mathrm{B}}^2(i)=0$, $g_{\mathrm{H}}(i)$ 达到极小且等于 -1.

将式 (7.2.4) 和式 (7.2.5) 与多维正态条件概率密度的贝叶斯方法的判别规则式 (2.3.3) 和式 (2.3.4) 对比, 两者是非常相似的, 不过前者的判别函数比后者的判别函数少了两项. 但是当训练样本集的样本数确定之后, 缺少的两项都是常数. 这样, 当我们利用式 (7.2.6)~(7.2.8) 作为 $\boldsymbol{H}$ 矩阵判别方法的判别规则时, 它与多维正态条件概率密度的贝叶斯方法的判别规则式 (2.3.3) 和式 (2.3.4) 是等价的. 因此 $\boldsymbol{H}$ 矩阵判别方法实际上是两类问题的多维正态条件概率密度的贝叶斯判别方法.

$\boldsymbol{H}$ 矩阵判别方法的优点是算法简单、明了, 但是它的前提是信号/本底样本集服从多维正态分布, 这限制了它的适用范围. 即使符合这一前提, Fisher 判别方法的性能也与之相当或更优. 由于这些因素, $\boldsymbol{H}$ 矩阵判别方法在实际中使用较少.

7.3　函数判别分析

分类器的实质在于确定一个最佳判别函数, 利用它来确定未知样本的类别.

对于线性不可分的样本集, 神经网络、决策树和下面即将介绍的支持向量机提供了非线性关联数据判别的近似解, 如果所选择的分类器结构足够灵活, 训练样本统计量足够大, 原则上可达到任意精度. 但是一般说来, 这些方法比较复杂, 解析程度很差, 问题的求解过程缺乏 “透明性”.

对于线性可分的样本集, 利用第三章讨论的线性判别方法可以实现未知样本类别的正确判别. 用以决定样本类别的判别函数 $g(\boldsymbol{x})$ 为特征向量 $\boldsymbol{x}$ 的一次 (线性) 函数. 例如对于两类问题, Fisher 方法的判别函数 $g(\boldsymbol{x})$ 为式 (3.2.21) 所示

$$g(\boldsymbol{x})=\boldsymbol{w}^{*\mathrm{T}}\boldsymbol{x}-y_0$$

它为线性关联的数据提供了问题的解析解. 一般地, 对于线性不可分的数据, 找不到这样的判别函数. 在实际问题中, 往往事先无法知道样本集是否线性可分. 因此希望能找到一种同时适用于样本集线性可分和线性不可分情况的算法. 这种算法, 对于线性可分问题应当对两类样本集的所有样本能正确地分类; 而对于线性不可分问题, 则能得到一个被错分的样本数达到极小的解. 上述准则称为最小错分样本数准则. 3.4 和 3.5 节给出了两种特定的符合该准则的算法, 它们的共同点是利用规范化增广样本向量 $\boldsymbol{y}_m$ 和权向量 $\boldsymbol{v}$ 构建一个准则函数 $J(\boldsymbol{v})$, $J(\boldsymbol{v})$ 取极小值或极大值时的 $\boldsymbol{v}$ 为问题的最优解 $\boldsymbol{v}^*$. 这里准则函数 $J(\boldsymbol{v})$ 有特定的形式 (见 3.4 和 3.5 节的讨论). 这类方法的优点是解题方法相对简单, 而且问题的求解过程具有 "透明性". 不过对于具有复杂非线性关联的数据样本, 其判别性能变差.

文献 [45] 中讨论的函数判别分析 (function discriminant analysis, FDA) 与最小错分样本数准则的判别函数法是类似的. 它的基本思想如下: 设用以决定样本类别的判别函数为 $g(\boldsymbol{x},\boldsymbol{a})$, 它是特征向量 $\boldsymbol{x}$ 和可调参数向量 $\boldsymbol{a}=(a_1,a_2,\cdots,a_m)^{\mathrm{T}}$ 的函数. FDA 法根据类别已知的训练样本集进行训练, 使得对于信号样本, 判别函数的值尽可能接近 1, 本底样本的判别函数的值尽可能接近 0. 定义估计量 $Q(\boldsymbol{a})$:

$$Q(\boldsymbol{a})=\frac{1}{w_{\mathrm{S}}}\sum_{i=1}^{N_{\mathrm{S}}}w_i\left(g(\boldsymbol{x}_i,\boldsymbol{a})-1\right)^2+\frac{1}{w_{\mathrm{B}}}\sum_{i=1}^{N_{\mathrm{B}}}w_i g(\boldsymbol{x}_i,\boldsymbol{a})^2 \tag{7.3.1}$$

其中, N_{S} 和 N_{B} 为训练样本集中的信号样本和本底样本数, $N=N_{\mathrm{S}}+N_{\mathrm{B}}$, w_i 为样本 i 的权值, w_{S} 和 w_{B} 为训练样本集中的信号样本和本底样本权值之和, 即

$$w_{\mathrm{S}}=\sum_{i=1}^{N_{\mathrm{S}}}w_i,\qquad w_B=\sum_{i=1}^{N_B}w_i. \tag{7.3.2}$$

一般情况下, 如果认为每个样本点的重要性是相等的, 则对每一个样本点赋予同样的权重, 这时有

$$\begin{cases} w_i=\dfrac{1}{N}, & i=1,2,\cdots,N; \\ w_{\mathrm{S}}=\dfrac{N_{\mathrm{S}}}{N}, & w_{\mathrm{B}}=\dfrac{N_{\mathrm{B}}}{N}. \end{cases} \tag{7.3.3}$$

如果每个样本点的抽取是不等概率的, 那么, 每一个样本点的权重 w_i 可以不同.

由 $Q(\boldsymbol{a})$ 定义可知, 它是 N 个训练样本的判别函数 $g(\boldsymbol{x},\boldsymbol{a})$ 值与其预期值 (信号样本为 1, 本底样本为 0) 的离差的加权平方和. $Q(\boldsymbol{a})$ 的大小是判别函数 $g(\boldsymbol{x},\boldsymbol{a})$ 保真性的度量, $Q(\boldsymbol{a})$ 越接近于 0, 样本的错分率越小. 因此 $Q(\boldsymbol{a})$ 的极小值对应的可调参数向量 $\boldsymbol{a}$ 的值 $\boldsymbol{a}^*$ 即为问题的解:

$$Q_{\min}(\boldsymbol{a})=Q(\boldsymbol{a}^*). \tag{7.3.4}$$

$Q(\boldsymbol{a})$ 的极小值可通过现成的极小化程序包 (如 MINUIT[48]) 求解. 这样, 给定阈值 $0 < g_{\text{th}} < 1$, 对于任意类别未知的样本 $\boldsymbol{x}'$, 决策规则为

$$\begin{aligned} &g(\boldsymbol{x}', \boldsymbol{a}^*) \geqslant g_{\text{th}}, \qquad \boldsymbol{x}' \text{ 判为信号;} \\ &g(\boldsymbol{x}', \boldsymbol{a}^*) < g_{\text{th}}, \qquad \boldsymbol{x}' \text{判为本底.} \end{aligned} \tag{7.3.5}$$

显然 $Q(\boldsymbol{a})$ 的极小值 $Q_{\min}(\boldsymbol{a}) = Q(\boldsymbol{a}^*)$ 的大小取决于判别函数 $g(\boldsymbol{x}, \boldsymbol{a})$ 的选择. $Q_{\min}(\boldsymbol{a})$ 越接近于 0 对应的判别函数 $g(\boldsymbol{x}, \boldsymbol{a})$ 具有更强的判别性能. 但是文献 [45] 并没有给出确定判别函数 $g(\boldsymbol{x}, \boldsymbol{a})$ 形式和可调参数向量 $\boldsymbol{a}$ 的分量个数的方法, 这些是需要研究者根据自身对于问题的了解和经验加以确定的.

但是, 我们可以从 Fisher 方法对于判别函数的确定方法得到判别函数 $g(\boldsymbol{x}, \boldsymbol{a})$ 形式的启示. Fisher 方法的判别函数为, 对于任意类别未知的样本 $\boldsymbol{x}'$, 决策规则为式 (3.2.21) 所示

$$\begin{aligned} &g(\boldsymbol{x}') = \boldsymbol{w}^{*\text{T}}\boldsymbol{x}' - y_0 \geqslant 0, \qquad \boldsymbol{x}' \text{ 判为信号;} \\ &g(\boldsymbol{x}') = \boldsymbol{w}^{*\text{T}}\boldsymbol{x}' - y_0 < 0, \qquad \boldsymbol{x}' \text{ 判为本底.} \end{aligned}$$

它们可以改写为

$$\begin{cases} g_{\text{F}}(\boldsymbol{x}') \equiv \boldsymbol{w}^{*\text{T}}\boldsymbol{x}' \cdot \dfrac{g_{\text{th}}}{y_0} \geqslant g_{\text{th}}, & \boldsymbol{x}' \text{ 判为信号;} \\ g_{\text{F}}(\boldsymbol{x}') \equiv \boldsymbol{w}^{*\text{T}}\boldsymbol{x}' \cdot \dfrac{g_{\text{th}}}{y_0} < g_{\text{th}}, & \boldsymbol{x}' \text{ 判为本底.} \end{cases} \tag{7.3.6}$$

式 (7.3.6) 与式 (7.3.5) 有相似的形式. 这种相似性提示我们, 可以将 $\boldsymbol{x}'$ 的线性函数 $g_{\text{F}}(\boldsymbol{x}')$ 作为判别函数 $g(\boldsymbol{x}, \boldsymbol{a})$ 的线性部分的近似, 再加上若干个非线性项, 应当就是 $g(\boldsymbol{x}, \boldsymbol{a})$ 的比较适当的形式. 至于非线性项的多少和最高幂次的大小, 则需根据研究者对于问题的了解和经验通过试验加以确定. 对于非线性关联的训练样本集, 检查训练样本集特征向量各变量之间的样本协方差矩阵可给出各变量之间关联强度的信息, 检查各变量之间等概率包络面的形状可给出关联幂次的信息, 从而有助于决定关联项的多少和幂次.

对于关联不太复杂的数据样本, 采用多项式函数作为判别函数 $g(\boldsymbol{x}, \boldsymbol{a})$ 通常是不错的选择. 例如对于特征向量有三个分量、考虑到二次幂的多项式函数, 判别函数 $g(\boldsymbol{x}, \boldsymbol{a})$ 有如下的形式:

$$\begin{aligned} g(\boldsymbol{x}, \boldsymbol{a}) = {} & a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3 + a_4 x_1 x_2 + a_5 x_1 x_3 + a_6 x_2 x_3 \\ & + a_7 x_1^2 + a_8 x_2^2 + a_9 x_3^2. \end{aligned}$$

这里有 10 个待定参数 $\boldsymbol{a} = \{a_0, a_1, \cdots, a_9\}$. 如果对数据样本的关联有所了解, 知道某些变量之间的关联系数很小, 则可以略去相应的待定参数, 这样可以减小计算量. 对于特征变量之间比较复杂的关联, 比如指数关联、对数关联或其他更复杂函数形式的关联, 如果研究者通过对样本分布的研究已经有明确的结论, 也可以在判别函数中加上对应的关联项.

函数判别分析方法中, 判别函数具有解析形式, 因而解题方法相对简单, 而且问题的求解过程具有 "透明性", 算法易于跟踪和调整. 由于函数判别分析可以包含非线性关联项, 因此对于存在非线性关联的数据样本, 其判别性能应该优于最小错分样本数准则的线性判别函数法. 对于具有复杂非线性关联的数据样本, 其判别性能取决于判别函数的 "保真性". 由于复杂非线性关联一般来说很难用解析表式加以精确描述, 因此, 对于这类数据样本, 其判别性能一般来说不如神经网络、决策树和下面即将介绍的支持向量机.

7.4 支持向量机

传统的统计模式识别方法都是在样本数量足够大的前提下进行研究的, 只有在样本数趋向无穷大时其性能才有理论上的保证. V.N.Vapnik 等人早在 20 世纪 60 年代就开始研究有限样本情况下的机器学习问题[49,50]. 直到 90 年代中才形成一个较完善的理论体系 —— 统计学习理论 (statistical learning theory, SLT), 为研究有限样本量情况下的统计模式识别建立了一个理论框架[50,51]. 1992~1995 年间[51~53], 在统计学习理论的基础上, 发展了一种新的模式识别方法 —— 支持向量机 (support vector machine, SVM), 在解决小样本、非线性及高维模式识别问题中表现出许多特有的优势. 统计学习理论和支持向量机已经成为国际上机器学习领域新的研究热点.

统计模式识别问题可以看作一个更广义问题 —— 基于数据的机器学习问题 —— 的特例. 基于数据的机器学习问题是现代智能技术中十分重要的一个方面, 主要研究如何从观测数据 (样本) 出发求得尚不能通过原理分析得到的规律, 利用这些规律再对未来数据或无法观测的数据进行预测. 当我们把要研究的规律抽象成分类关系时, 这种机器学习问题就是模式识别.

7.4.1 最优分类面

支持向量机是统计学习理论中最实用的部分, 其核心思想是将结构风险最小化原则引入分类方法之中.

SVM 方法是从线性可分情形下的最优分类面 (optimal hyperplane) 问题引出的. 本节的讨论中, 假定样本分为信号和本底两个类别, 并首先讨论线性可分的

情形. 考虑图 7.10 所示的两类线性可分的样本, 图中的实心和空心点分别表示两类的训练样本,H 为把两类样本正确无误地分开的分类线, H_1,H_2 分别为过两类样本离分类线最近的点且平行于 H 的直线, H_1,H_2 间的距离称为两类的分类间隙 (margin). 所谓最优分类线就是要求分类线不但能将两类样本正确无误地分开, 而且能使两类的分类间隙最大. 前一要求是为了保证经验风险最小, 而后一要求是使真实风险最小. 推广到高维空间, 最优分类线就成为最优分类面.

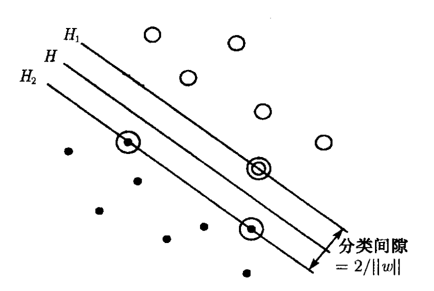


图 7.10 最优分类面示意图

设样本集包含 N 个样本: $\boldsymbol{x}_i, i=1,2,\cdots,N$, $\boldsymbol{x}_i \in R^n$, 样本的类别用 $y_i \in \{+1,-1\}$ 表示. 由式 (3.1.3) 知两类情况下线性判别函数的一般表式为 $g(\boldsymbol{x}) = \boldsymbol{w}\cdot\boldsymbol{x}+b$, 当 $g(\boldsymbol{x}_i)>0$, 样本类别 $y_i=+1$; 当 $g(\boldsymbol{x}_i)<0$, $y_i=-1$. 分类面方程为

$$g(\boldsymbol{x}) = \boldsymbol{w}\cdot\boldsymbol{x}+b=0. \tag{7.4.1}$$

可以适当选择 $\boldsymbol{w}$ 和 b 的乘因子, 使得两类的所有样本都满足 $|g(\boldsymbol{x})| \geqslant 1$, 即使离分类面最近的样本满足 $|g(\boldsymbol{x})|=1$, 这样, 两类的分类间隙 (margin) 就等于 $2/\|\boldsymbol{w}\|$ (参见 3.1 节 "线性判别函数"). 因此使分类间隙最大等价于使 $\|\boldsymbol{w}\|$ (或 $\|\boldsymbol{w}\|^2$) 最小; 而要求分类面对所有样本分类正确就是要求满足

$$y_i(\boldsymbol{w}\cdot\boldsymbol{x}_i+b)-1 \geqslant 0, \qquad i=1,2,\cdots,N. \tag{7.4.2}$$

因此, 满足上述条件并使 $\|\boldsymbol{w}\|^2$ 最小的分类面就是最优分类面. 过两类样本离分类面最近的点且平行于最优分类面 H 的超平面 H_1,H_2 上的训练样本就是式 (7.4.2) 中使等号成立的那些样本, 它们被称为支持向量 (support vectors), 因为它们支撑了最优分类面. 在图 7.10 中它们用圆圈标出的点所示.

下面来讨论如何求得最优分类面. 根据上面的讨论, 最优分类面的求解可以表示为在条件式 (7.4.2) 的约束下, 求函数

$$\varphi(\boldsymbol{w}) = \|\boldsymbol{w}\|^2/2 = \boldsymbol{w}\cdot\boldsymbol{w}/2 \tag{7.4.3}$$

的极小值问题. 为此, 定义 Lagrange 函数

$$L(\boldsymbol{w},b,\boldsymbol{\alpha})=\frac{1}{2}(\boldsymbol{w}\cdot\boldsymbol{w})-\sum_{i=1}^{N}\alpha_i\left[y_i\left(\boldsymbol{w}\cdot\boldsymbol{x}_i+b\right)-1\right], \tag{7.4.4}$$

其中, $\alpha_i>0$ 为 Lagrange 系数.

我们的问题化为求 Lagrange 函数对 $\boldsymbol{w}$ 和 b 的极小值, 并同时满足 Lagrange 函数对于所有的 α_i 的导数等于 0, 以及 $\alpha_i>0$. 将式 (7.4.4) 分别对 $\boldsymbol{w}$ 和 b 求导并令它们等于 0, 得到

$$\boldsymbol{w}=\sum_{i=1}^{N}\alpha_i y_i\boldsymbol{x}_i,$$

$$\sum_{i=1}^{N}y_i\alpha_i=0. \tag{7.4.5}$$

Wolf 对偶问题告诉我们[54], 在 $L(\boldsymbol{w},b,\boldsymbol{\alpha})$ 函数对 $\boldsymbol{w}$ 和 b 的导数等于 0, 并满足约束 $\alpha_i>0$ 的条件下, $L(\boldsymbol{w},b,\boldsymbol{\alpha})$ 对于 α_i 的极大值解与 $L(\boldsymbol{w},b,\boldsymbol{\alpha})$ 对 $\boldsymbol{w}$ 和 b 的极小值解将得到同样的解 $\boldsymbol{w}^*$, b^* 和 α^*. 将式 (7.4.5) 代入式 (7.4.4), 我们的问题转化为在满足约束 $\alpha_i>0(i=1,2,\cdots,N)$ 的条件下对 α_i 求解下列称为对偶函数的最大值

$$Q(\alpha)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j(\boldsymbol{x}_i\cdot\boldsymbol{x}_j). \tag{7.4.6}$$

一般, $Q(\boldsymbol{\alpha})$ 的最大值解 α_i^* 需用训练样本特征向量 $\boldsymbol{x}_i(i=1,2,\cdots,N)$ 及其类别 y_i 通过数值方法求得. 若 α_i^* 为最优解, 则有

$$\boldsymbol{w}^*=\sum_{i=1}^{N}\alpha_i^* y_i\boldsymbol{x}_i, \tag{7.4.7}$$

即最优分类面的权系数向量是训练样本向量的线性组合.

这是一个不等式约束下的二次函数极值问题, 存在唯一解. 且根据 Karush-Kühn-Tucker 条件[54], 这个优化问题的解须满足

$$\alpha_i\left[y_i\left(\boldsymbol{w}\cdot\boldsymbol{x}_i+b\right)-1\right]=0,\qquad i=1,2,\cdots,N. \tag{7.4.8}$$

因此, 对照式 (7.4.2)$y_i(\boldsymbol{w}\cdot\boldsymbol{x}_i+b)-1\geqslant 0,\ i=1,2,\cdots,N$ 可知, 只有该式等号成立的样本, 即支持向量对应的 α_i^* 不为 0, 而其他所有样本对应的 α_i^* 须等于 0.

求解上述问题后得到的最优分类函数是

$$f(\boldsymbol{x}) = \operatorname{sgn}(\boldsymbol{w}^* \cdot \boldsymbol{x} + b^*) = \operatorname{sgn}\left\{\sum_{i=1}^{N} \alpha_i^* y_i (\boldsymbol{x}_i \cdot \boldsymbol{x}) + b^*\right\}. \tag{7.4.9}$$

注意, 由于非支持向量对应的 α_i^* 均等于 0, 式 (7.4.7) 和 (7.4.9) 中的求和实际上只对少数支持向量进行. b^* 是分类的阈值, 可以由任意一个支持向量通过式 (7.4.8) 求得

$$b^* = y_i^{-1} - \boldsymbol{w}^* \cdot \boldsymbol{x}_i. \tag{7.4.10}$$

对于实际应用, 取所有支持向量计算得到的 b^* 值的平均作为阈值更为安全. 这样, 对于任意未知待分类样本 $\boldsymbol{x}$, 就可由式 (7.4.9) 求得其类别 $y = f(\boldsymbol{x})$.

7.4.2 广义最优分类面

最优分类面是在线性可分的前提下讨论的. 当样本线性不可分, 即某些训练样本不能满足式 (7.4.2) 规定的条件, 可将约束条件修改为

$$y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) - 1 + \xi_i \geqslant 0, \qquad \xi_i \geqslant 0, i = 1, 2, \cdots, N. \tag{7.4.11}$$

其中, ξ_i 称为 "松弛量" (slack variable). 当样本 $\boldsymbol{x}_i$ 落在分类面 H: $g(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b = 0$ 上时有 $\xi_i = 1$, 故当样本 $\boldsymbol{x}_i$ 被分类面 H 错分时必有 $\xi_i > 1$. 于是量 $\sum_i \xi_i$ 可视为 N 个训练样本中被错分的样本数的上界. 现在, 样本线性不可分情形下的广义最优分类面问题可演化为在条件式 (7.4.11) 的约束下求函数

$$\varphi(\boldsymbol{w}, \boldsymbol{\xi}) = \frac{1}{2}\boldsymbol{w} \cdot \boldsymbol{w} + C\sum_{i=1}^{N} \xi_i \tag{7.4.12}$$

的极小值. 式中 C 称为费用参数 (cost parameter), 它是一个给定的常数, 起着控制对错分样本惩罚程度的作用. C 越大, 对错分样本的惩罚程度越高. 因此, Lagrange 函数为

$$L(\boldsymbol{w}, b, \alpha, \xi) = \frac{1}{2}(\boldsymbol{w} \cdot \boldsymbol{w}) + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i [y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{N} \mu_i \xi_i, \tag{7.4.13}$$

其中, $\mu_i > 0$ 是为了保证 $\xi_i > 0$ 而引入的 Lagrange 系数. 将式 (7.4.13) 分别对 $\boldsymbol{w}$ 和 b 求导并令它们等于 0, 得到与式 (7.4.5) 同样的结果,

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x}_i,$$

$$\sum_{i=1}^{N} y_i\alpha_i = 0.$$

将式 (7.4.13) 对 ξ_i 求导并令它们等于 0, 得到

$$\alpha_i + \mu_i = C \tag{7.4.14}$$

因为 $\alpha_i > 0$, $\mu_i > 0$, 所以式 (7.4.14) 表示 C 是 α_i, μ_i 的上界.

用与求解最优分类面同样的 Wolf 对偶问题方法求解这一优化问题, 同样得到一个二次函数的极值问题, 其结果与线性可分情形下得到的式 (7.4.5)~(7.4.7) 和 (7.4.9) 几乎完全相同, 只是在求解式 (7.4.6) 的对偶函数 $Q(\boldsymbol{\alpha})$ 对 α_i 的最大值时要求满足

$$C \geqslant \alpha_i \geqslant 0, \qquad i = 1, 2, \cdots, N \tag{7.4.15}$$

来代替原约束条件 $\alpha_i > 0\,(i = 1, 2, \cdots, N)$ 即可[55]. 因此, 对于样本线性不可分的情形, 广义最优分类面的解仍由式 (7.4.7) 表示, 对任意未知待分类样本 $\boldsymbol{x}$, 仍由式 (7.4.9) 求得其类别 $y = f(\boldsymbol{x})$. 同样, 式中的求和实际上只对少数支持向量进行. 式中的分类阈值 b^* 根据如下方法计算. 由于上述优化问题须满足 Karush-Kühn-Tucker 补充条件:

$$\alpha_i \left[y_i \left(\boldsymbol{w} \cdot \boldsymbol{x}_i + b \right) - 1 + \xi_i \right] = 0, \qquad i = 1, 2, \cdots, N. \tag{7.4.16}$$

$$\mu_i \xi_i = 0, \qquad i = 1, 2, \cdots, N. \tag{7.4.17}$$

由式 (7.4.14) 和式 (7.4.17) 知, 对于满足 $C > \alpha_i > 0$ 的任意样本有 $\mu_i > 0, \xi_i = 0$, 因此据式 (7.4.16) 知, 选择满足 $C > \alpha_i > 0$ 的任意样本 $\boldsymbol{x}_i$, 分类阈值 b^* 可按下式计算:

$$b^* = y_i^{-1} - \boldsymbol{w}^* \cdot \boldsymbol{x}_i. \tag{7.4.18}$$

对于实际应用, 取所有满足 $C > \alpha_i > 0$ 的训练样本计算得到的 b^* 值的平均作为阈值更为安全.

7.4.3 支持向量机

上面讨论的最优分类面和广义最优分类面问题中, 其分类判别函数式 (7.4.9) 中只包含待分类样本 $\boldsymbol{x}$ 与训练样本中的支持向量的内积运算 $\boldsymbol{x}_i \cdot \boldsymbol{x}$. 可见, 要解决一个特征空间中的最优线性分类问题, 只需要知道这个空间中的内积运算即可.

回顾 3.1 节中的广义线性判别函数问题, 如果一个问题在其定义的空间中不是线性可分的, 这时可以考虑构造新的特征向量, 把问题转换到一个新的、更高维的空间中, 在那里可以用线性判别函数实现原空间中的非线性判别. 比如构造

$\boldsymbol{y}=[1\ x\ x^2]^{\mathrm{T}}$, 就可以用线性函数 $g(\boldsymbol{y})=\boldsymbol{v}^{\mathrm{T}}\boldsymbol{y}$ 实现 $g(\boldsymbol{x})=c_0+c_1x+c_2x^2$ 的非线性判别问题, 其中广义权向量为 $\boldsymbol{v}=[c_0\ c_1\ c_2]^{\mathrm{T}}$. 实际上, 一般来说, 对于任意高次的判别函数, 都可以通过适当的变换转化为更高维空间中的线性判别函数来处理. 这时变换后的空间中的线性判别函数称为广义线性判别函数.

按照广义线性判别函数的思路, 要解决一个非线性问题, 可以设法将它通过非线性变换 (用函数 φ 表示) 转化为另一个更高维空间 G 中的线性问题, 在这个空间中求最优或广义最优分类面, 这时原空间中的内积 $\boldsymbol{x}_i\cdot\boldsymbol{x}$ 在 G 空间中变为 $\varphi(\boldsymbol{x}_i)\cdot\varphi(\boldsymbol{x})$. 统计学习理论指出[52], 根据 Hilbert-Schmidt 原理, 只要满足 Mercer 条件, 点积 $\varphi(\boldsymbol{x}_i)\cdot\varphi(\boldsymbol{x})$ 可以用核函数 (Kernel function) $K(\boldsymbol{x},\boldsymbol{x}_i)$ 作为近似. Mercer 条件指的是, 对于任意的对称函数 $K(\boldsymbol{x},\boldsymbol{x}')$, 它是某个特征空间中的内积运算的充分必要条件是, 对于任意的 $\varphi(\boldsymbol{x})\neq 0$ 且 $\int\varphi^2(\boldsymbol{x})\mathrm{d}\boldsymbol{x}<\infty$, 有

$$\iint K(\boldsymbol{x},\boldsymbol{x}')\varphi(\boldsymbol{x})\varphi(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}'>0. \tag{7.4.19}$$

这一条件通常不难满足. 这样我们就可以避免变换函数 $\varphi(\boldsymbol{x})$ 的计算, 因为 $\varphi(\boldsymbol{x})$ 的严格表式难以从训练数据导出.

在这种情况下, 式 (7.4.6) 的优化函数变为

$$Q(\boldsymbol{\alpha})=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i,\boldsymbol{x}_j), \tag{7.4.20}$$

而相应的判别函数式 (7.4.9) 也应变为

$$f(\boldsymbol{x})=\mathrm{sgn}\,(\boldsymbol{w}^*\cdot\boldsymbol{x}+b^*)=\mathrm{sgn}\left\{\sum_{i=1}^{N}\alpha_i^* y_i K(\boldsymbol{x}_i,\boldsymbol{x})+b^*\right\}. \tag{7.4.21}$$

算法的其他部分不变. 这就是支持向量机算法. 由于判别函数中只包含未知样本 $\boldsymbol{x}$ 与支持向量 $\boldsymbol{x}_i$ 的点积求和, 因此计算量取决于支持向量的个数.

支持向量机求得的分类函数形式上类似于一个神经网络, 其输出是若干中间层节点的线性组合, 每一个中间层节点对应于输入样本与一个支持向量的点积, 因此支持向量机也被称为支持向量网络, 如图 7.11 所示.

利用不同的核函数将导致不同的支持向量机算法, 目前研究的核函数主要有三类, 它们与已有的方法有对应关系.

(1) 多项式核函数

$$K(\boldsymbol{x},\boldsymbol{x}_i)=(\boldsymbol{x}\cdot\boldsymbol{x}_i+1)^q, \tag{7.4.22}$$

此时的支持向量机是一个 q 阶多项式分类器.

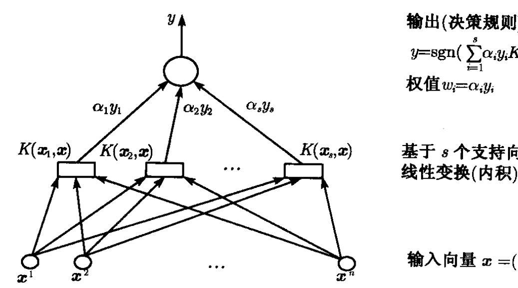


图 7.11 支持向量机示意图

(2) 高斯型核函数

$$K(\boldsymbol{x},\boldsymbol{x}_i)=\exp\left[-\frac{|\boldsymbol{x}-\boldsymbol{x}_i|^2}{2\sigma^2}\right], \tag{7.4.23}$$

此时的支持向量机是一种径向基函数分类器. 它与一般的径向基函数 (RBF) 方法的基本区别是, 这里每一个基函数的中心对应于一个支持向量, 它们以及输出权值都是由算法自动确定的.

(3) S 型核函数

$$K(\boldsymbol{x},\boldsymbol{x}_i)=\tanh\left[\kappa(\boldsymbol{x}\cdot\boldsymbol{x}_i)+\theta\right], \tag{7.4.24}$$

此时的支持向量机是一个多层感知器神经网络, 但网络的权值和网络隐层节点的数目都是由算法自动确定的.

式 (7.4.22)~(7.4.24) 中的 q,σ,κ,θ 都是可选常数. 应当指出, 在这三种常用的核函数中, 前两种满足 Mercer 条件, 而 S 型核函数只对某些特定 κ,θ 值才满足 Mercer 条件.

图 7.12 是利用 q=2 的多项式核函数的支持向量机算法对两类样本的分类结

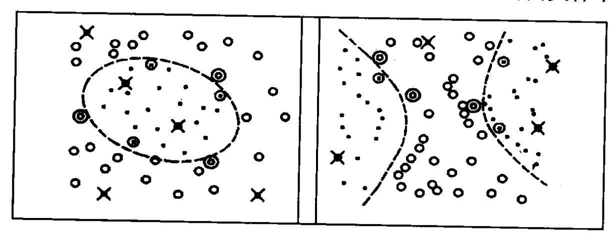


图 7.12 q=2 的多项式核函数支持向量机分类结果

果示意图. 图中小圆圈和黑点代表两类样本点, 虚线画出了 $q=2$ 的多项式核函数求得的支持向量机分类线, 划圆圈的样本点是求得的支持向量, 划叉的样本点表示错分的样本.

关于支持向量机的错分率, 有如下结论: 如果一组训练样本能被一个最优分类面或广义最优分类面分开, 则对于测试样本分类错误率的期望值的上界等于训练样本集中支持向量个数 N_{SV} 的平均值占训练样本总数 N 的比例, 即

$$E\left[\varepsilon(e)\right] \leqslant \frac{E\left[N_{\mathrm{SV}}\right]}{N-1}. \tag{7.4.25}$$

因此, 当支持向量个数 N_{SV} 很小时, 错分率也很小. 而且, 错分率与核函数的选择关系不大. 此外, 在满足 Mercer 条件的情形下, 相应的最优化问题是一个凹二次项的极小化问题, 其解收敛于全局极小, 这一点比可能收敛于局部极小的神经网络要来得优越.

对于三种常用的核函数的支持向量机的对比研究表明, 不同核函数的支持向量机其性能是相近的, 不像神经网络那样十分依赖于模型的选择, 此外三种核函数求得的支持向量个数只是训练样本总数的很小一部分, 而且三组支持向量中大部分是重合的. 当然, 这些优点只是来自于具体的对比研究, 它们是不是支持向量机的普遍性质有待于进一步的理论研究.

第八章　不同判别方法的比较

8.1　不同判别方法的特点

对于一个特定的分类问题, 利用何种判别方法能达到最优的分类效果, 主要需考虑两方面的因素: 待分类问题的性质和复杂程度, 以及所采用的判别方法的适用范围和性能. 为此, 当我们面临一个分类问题时, 必须选择一个恰当的判别方法达到问题的最优解. 这就要求对不同的判别方法的优缺点有所对比.

评价一种判别方法的优良程度, 大体需要考虑以下几方面的因素 :

(1) 适用问题的范围.

(2) 方法涉及的理论的简单性和准确性.

(3) 判别性能, 即判别效率和误判率.

(4) 编程的简单程度, 计算速度和计算量.

一个最优的判别方案必须针对特定问题的特定要求综合考虑上述因素. 例如对于数据量小的样本集的分类问题, 可以降低对因素 (4) 的要求而提高对因素 (3) 的要求; 而对于大样本集的分类问题, 必须同时兼顾因素 (3) 和 (4) 的要求, 等等.

下面, 对于前面各章讨论过的各种判别方法作一概略的评述.

1. **贝叶斯决策**

贝叶斯决策分类的重要前提是, 要求对应于各类别 ω_i 出现的先验概率 $\pi(\omega_i)$ 和样本 $\boldsymbol{x} \in \omega_i$ 时的条件概率密度 $p(\boldsymbol{x}|\omega_i)$ 都是已知的. 在满足这两个条件的情形下, 它适用于任何问题的分类, 并且具有理论上的简单和准确性. 它的计算简单, 计算量不大. 基于最小错误率的贝叶斯决策使平均错误率达到最小, 即它的分类错误率在所有可能的分类器中是最小的, 因而就判别性能而言, 贝叶斯决策具有理论上的最优性能. 但贝叶斯决策分类的重要前提, 即要求样本 $\boldsymbol{x} \in \omega_i$ 的条件概率密度 $p(\boldsymbol{x}|\omega_i)$ 都为已知, 在实际问题中通常是不满足的. 因此必须首先对类条件概率密度 $p(\boldsymbol{x}|\omega_i)$ 进行估计. 这需要统计学的一套复杂的方法. 一种常用的类条件概率密度是多维正态分布假设, 在使用它时应注意该假设在物理上的合理性, 或者先进行假设检验, 否则会导致结果的不可靠.

2. **线性判别方法**

线性判别方法利用样本的线性函数作为样本类别的判别函数, 方法简单, 容易实现, 计算量和数据存储量小, 因而是实际应用中常用的方法之一. 对于线性可分

的数据样本的分类问题, 基于线性判别的分类器 (Fisher 线性判别, 感知准则函数判别) 具有最佳的判别性能. 对于两类而且线性可分的数据, 线性判别分类器能够对全部样本正确分类, 应该作为分类方法的首选. 对于多类问题, 虽然两类问题的准则和算法原则上可以推广到多类情况, 但是计算相当复杂, 因而编程比较复杂. 对于线性不可分的数据样本的分类问题, 虽然最小错分样本数准则函数和最小平方误差准则函数方法也能在准则函数最优化的意义下减小错分率, 但与贝叶斯决策或非线性判别方法相比, 错分率一般是比较大的, 因而不宜采用.

3. *决策树方法*

最简单的二元决策树方法 —— 超长方体分割法, 具有思路清晰、简明和物理的直观性, 程序设计和调试特别简单, 计算速度快等优点, 因而在实验数据分析中有广泛的应用. 但其缺点是, 当信号和本底样本的条件概率密度函数相互重叠而不相分离时, 或数据存在非线性关联时, 信号样本的判选效率下降, 错判率增加. 此外, 用来区分类信号区/类本底区的阈值向量 $\boldsymbol{x}^{\rm th}$ 的最优值的确定比较困难, 多少个变量用于判选能达到分类器性能/计算时间的最优组合比较难以确定. 对样本数据首先进行主成分分析得到新特征向量数据, 然后用超长方体分割法进行信号和本底样本的分类, 能在一定程度上提高分类器的性能, 因而值得推荐.

一般的二元决策树方法通过某种优化步骤, 在每一节点中选择区分信号和本底能力最强的那个变量, 从而使其判别性能较之超长方体分割法有所提高. 但是, 确定每个节点的最佳 (变量 + 阈值) 组合, 确定最佳的决策树长度, 亦即避免过度训练, 仍是一个困难问题 ; 分类器性能对于训练样本集的统计涨落具有不稳定性的问题亦难以解决.

决策树林法通过构造多棵决策树, 经过加权后结合成一个分类器, 它使得样本分类的正确性对训练样本集的统计涨落不敏感. 决策树林法对非线性关联数据有很强的分类判别能力. 虽然它的计算的复杂性和计算量较之单个决策树有明显的增加, 避免过度训练问题需要解决, 但是与人工神经网络相比较, 决策树林法的设计仍是比较简单的, 计算量是相对小的. 由于这种简单性, 决策树林法的理论最优性能略逊于人工神经网络, 但对于训练样本量不是特别大, 而数据存在复杂相关性的情形, 决策树林法的性能优于其他方法.

4. *人工神经网络*

人工神经网络也许是所讨论过的方法中对非线性复杂关联数据具有最强判别能力的一种方法. 这可能是它的最大优势. 它的基本思想似乎是简单的, 即将 n 维空间的特征向量转化为一维输出变量, 该输出变量对于信号和本底样本是明显分离的, 因而易于加以判别. 然而这种基本思想的实现方式在人工神经网络中缺乏物理的直观性. 利用 Sigmoid 函数作为变换函数的三层 BP 网络可以以任意精度逼近

任意连续函数. 也就是说, 三层 BP 网络原则上可以解决任意非线性的分类问题. 其缺点是: 隐含层的节点数的确定缺乏理论指导和有效的方法, 可能陷入局部极小而得不到全局极小点, 决定收敛速度的权值修正系数 (学习率)η 的确定依赖于尝试和经验. 对于 BP 网络学习算法的改进方案 (如全局误差极小化方法, 引入惯性修正项, 用变步长法代替一阶梯度法寻优) 一定程度上改善了对于尝试和经验的依赖, 提高了计算速度, 但对于前两个缺点的克服帮助不大.

Hopfield 网络是一种全连接型反馈网络, 连续型 Hopfield 神经网络中各神经元采用并行方式工作, 所以在信息处理的并行性、联想性、实时性方面有更强的能力. 但是, 它与 BP 网络一样, 用某个目标函数的全局极小作为算法搜索和网络状态变化的依据, BP 网络的目标函数是误差函数, Hopfield 网络中是能量函数. 因此同样存在可能陷入局部极小而得不到全局极小的问题.

引入了模拟退火算法的 Boltzmann 机即 BM 网络比 BP 网络和 Hopfield 网络有更高的概率达到全局极小, 且这一算法具有很强的通用性, 特别是对复杂性较高、规模较大、对问题的有关知识了解较少的情况, 它具有明显的优越性. 但是, BM 网络学习规则中, 包含着其工作规则, 学习与反学习交替进行, 因此, 网络计算量大, 特别是当网络温度下降速度较慢时, 网络收敛过程缓慢, 这是制约 BM 网络算法应用的主要障碍.

不论哪种神经网络, 它的设计、训练都是比较复杂并且耗时的, 计算量和数据存储量很大, 并需要有足够统计量的训练样本集. 鉴于它的判别能力很强, 一般用于数据关联复杂、多类别的分类问题, 例如粒子物理实验数据分析中的粒子鉴别问题.

5. **近邻法**

最近邻法的决策思想简单而又直观, 对于任意待归类的样本 x, 判定它与离它欧氏距离最近的那个训练样本同类. 它的显著缺点是有较高的错判率. k 近邻法是最近邻法的一种推广. 对于任意待归类的样本 x, 取它的 k 个近邻训练样本, 这 k 个近邻样本中哪一个模式类的样本数量最多, 就把样本 x 判为哪一类. 这一做法减小了其错判率. 近邻法错误率介于 ε_B 和 $2\varepsilon_B$ 之间, 其中 ε_B 为贝叶斯决策错误率. 无论是近邻法还是 k 近邻法, 其基本思想和算法步骤都十分简单, 计算速度快, 其性能对于线性可分或不可分数据没有明显差别, 这使它成为常用的重要分类方法之一. 它的缺点是, 每次决策都要计算待识别样本 x 与全部训练样本之间的距离并进行比较. 当训练样本量 N 很大时, 存储量和计算量都较大. 上述的性能分析是渐近的平均结果, 即要求 $N \to \infty$, 这在实际场合是无法实现的, 实际错判率与预期值可能存在差别, 因此会产生较大的风险.

剪辑近邻法利用类别已知的训练集来估计错分率应该是较为准确的. 特别是

k 近邻剪辑法和重复剪辑近邻法当 $k \to \infty$ 时其错误率收敛于最优错误率 ε_{B}, 提高了信号样本的判选效率. 具有拒绝决策的 k 近邻法和剪辑近邻法则减小了决策出现高风险的概率, 减小了错误率. 这些计算都不算复杂, 而对分类器的性能有明显的提高, 因而推荐使用.

6. 概率密度估计量方法

所谓概率密度估计量方法, 其实质即是利用类别已知的训练样本集来求得类条件概率密度 $p(\boldsymbol{x}|\omega_i)$ 的估计 $\hat{p}(\boldsymbol{x}|\omega_i),(i=1,2,\cdots,c)$, 然后用贝叶斯判别方法来进行样本分类. 显然, 该方法的性能依赖于 $\hat{p}(\boldsymbol{x}|\omega_i)$ 对于类条件概率密度 $p(\boldsymbol{x}|\omega_i)$ 的逼近程度. 在投影似然比估计中, 特征向量 $\boldsymbol{x}=(x_1,\cdots,x_n)^{\mathrm{T}}$ 的 n 个变量考虑为互不关联, 这时类条件概率密度可以因子化为 n 个变量边沿概率密度的简单乘积. 这种方法固然计算简单, 计算量和存储量都不大, 但实际数据往往存在复杂的关联, 如果对这种关联数据使用投影似然比估计方法分类, 其错分率总是比较大, 并且具有某种不可控制性. 利用训练样本估计多维类条件概率密度的方法为 PDE-RS (PDE range search) 方法, 它的基本思想是利用 k_N 近邻方法估计多维类条件概率密度. 为了使得 $\hat{p}(\boldsymbol{x}|\omega_i)$ 能很好地逼近类条件概率密度 $p(\boldsymbol{x}|\omega_i)$, 需要很大数量的已知类别的训练样本集, 这使得 PDE-RS 方法的计算量和存储量都比较大, 计算速度较慢. 但是它的编程相对简单, 容易调试和追踪, 能处理复杂的非线性关联. 在特征变量维数不特别高、训练样本量足够大的情形下, 该分类器的性能具有竞争力, 即信号样本的选择效率较高, 错分率较小.

7. H 矩阵判别

H 矩阵判别方法实际上是两类问题的多维正态条件概率密度的贝叶斯判别方法. H 矩阵判别方法的优点是算法简单、明了, 但是它的前提是信号/本底样本集服从多维正态分布, 这限制了它的适用范围. 即使符合这一前提, Fisher 判别方法的性能也与之相当或更优. 由于这些因素, H 矩阵判别方法在实际中使用较少.

8. 函数判别分析

函数判别分析的基本思想是: 设用以决定样本类别的判别函数为 $g(\boldsymbol{x},\boldsymbol{a})$, 它是特征向量 $\boldsymbol{x}$ 和可调参数向量 $\boldsymbol{a}$ 的函数. FDA 法根据类别已知的训练样本集进行训练求得 $\boldsymbol{a}$ 的值, 使得对于信号样本, 判别函数的值尽可能接近 1, 本底样本的判别函数的值尽可能接近 0, 这样就实现了样本的分类. 对于线性不可分的数据, 一般找不到这样的判别函数, 目前的方法只是使被错分的样本数达到某种极小的解. 该方法的优点是判别函数具有解析形式, 解题方法相对简单, 而且问题的求解过程具有 “透明性”, 算法易于跟踪和调整. 由于函数判别分析可以包含非线性关联项, 因此对于存在非线性关联的数据样本, 其判别性能应该优于最小错分样本数准则的线性判别函数法. 对于具有复杂非线性关联的数据样本, 其判别性能取决于判别函数

的"保真性". 由于复杂非线性关联一般来说很难用解析表式加以精确描述, 因此, 对于这类数据样本, 其判别性能一般来说不如神经网络、决策树和支持向量机. 该方法的另一个明显缺点是判别函数 $g(\boldsymbol{x}, \boldsymbol{a})$ 的形式是未知的, 需要依靠使用者对于待分类样本集的分布有相当程度的了解后依靠经验来加以确定, 这在许多情况下是一件极为困难的任务. 因此 $g(\boldsymbol{x}, \boldsymbol{a})$ 的形式是针对特定问题的, 缺乏普适性. 一般只有对于关联不太复杂的数据样本, 可以用简单的函数 (如多项式函数) 作为判别函数 $g(\boldsymbol{x}, \boldsymbol{a})$, 这种情形下使用函数判别分析才是实际可行的.

9. **支持向量机**

支持向量机对样本进行分类的基本思想是利用全部类别已知的训练样本集中的一小部分样本 (其特征向量称为支持向量) 来建立一个超平面, 达到判别信号/本底的目的. 该方法需要利用核函数 (如多项式、Gauss 函数、Sigmoidal 函数等), 它的判别性能有赖于核函数形式及其参数的选择 (如 Gauss 函数的标准离差), 以及费用参数 C 的选择, 而且最佳选择一定程度上是问题依赖的. 分类器的训练时间大体上正比于 n^2N, 这里 n 是特征向量维数,N 是训练样本数, 因此计算量大体与 PDE-RS 方法、决策树林法、k 近邻法相若. 支持向量机方法的优点是方法中的可调参数少, 训练比较容易完成；对于复杂非线性关联数据的分类性能好, 可以与决策树林法、人工神经网络的判别性能相比拟.

综合上述讨论, 对于一个特定问题如何选择一个适当的判别方法可有如下的一般性考虑. 对于线性可分的数据样本, 应选择线性判别方法. 对于线性不可分的数据样本, 如果关联比较简单并且研究者对数据的分布的函数形式已有相当好的了解, 可以采用函数判别分析. 当数据存在非线性关联但特征变量维数不特别高、训练样本量足够大的情形下, 可采用 PDE-RS 方法. 对于一般的高维、非线性关联数据样本, 应采用 (剪辑)k 近邻法、决策树林法或支持向量机. 对于存在很复杂的非线性关联的高维数据, 错误率要求严苛的问题, 应考虑采用人工神经网络.

文献 [45] 对于各种判别方法的性能给出了表 8.1 所示的评价, 可供参考.

表 8.1 文献 [45] 对各种判别方法的性能评价

		超长方体分割	投影似然比	PDE-RS	k 近邻法	H 矩阵判别	Fisher	神经网络	决策树林法	支持向量机
性能	线性或无关联	+	++	+	+	+	++	++	+	+
	非线性关联	0	0	++	++	0	0	++	++	++
速度	训练方式	0	++	++	++	++	++	+	0	0
	应用方式	++	++	0	+	++	++	++	+	+
稳健性	过度训练	++	+	+	+	++	++	+	0	++

续表

	超长方体分割	投影似然比	PDE-RS	k 近邻法	H 矩阵判别	Fisher	神经网络	决策树林法	支持向量机
低判别力变量	++	+	0	0	++	++	+	++	+
维数灾难	0	++	0	0	++	++	+	+	
方法透明性	++	++	+	+	++	++	0	0	0

注: ++ 表示性能优良, + 表示性能一般, 0 表示性能差. 维数灾难表示当特征向量维数增高时, 训练样本统计量和运算时间的增加. 方法稳健性指对于过度训练和使用判别能力不强的变量的不敏感性.

8.2　多元统计分析程序包 TMVA 简介

针对高能物理实验中数据量浩大, 所寻找的信号事例可能相当稀少这一特点, 由一批数理统计学家和高能物理实验数据分析工作者合作, 将多种判别方法编写成易于选择和实行的计算程序, 并有机地总汇在一起, 以便于最大程度地挖掘数据包含的有助于事例类别判选的信息, 寻找适合所研究问题的最佳判别方法. 在高能物理领域中, 朝这个方向努力的初期工作是 BaBar 合作研究组 1998 年研发的 Cornelius 程序包[56]. 近期则有 StattPatternRecognition 程序包[57,58] 和 TMVA 程序包[45]. 这里仅对 TMVA 程序包作一简单介绍.

TMVA (toolkit for multivariate data analysis) 是一个多元统计分析的工具性程序包, 该程序包包含的判别方法包括：

超长方体分割法
总体概率密度的投影似然比估计
总体概率密度的多维概率密度估计 (PRD-RS)
k 近邻法
Fisher 判别
函数判别分析
H 矩阵判别
人工神经网络
支持向量机
决策树林法
Predictive learning via rule ensembles

它们几乎覆盖了本书讨论的大部分判别方法 (其中最后一种方法本书未加叙述). 因此 TMVA 也适用于一般的多元统计分析问题.

TMVA 已经集成进基于 C++ 语言的面向对象的数据分析系统 ROOT[59], 因而具有强大和方便的各种服务功能和友好的使用界面. 对所有这些判别方法, TMVA

可以完成分类器的训练、测试和性能评估, 从而便于使用者从中选择对自身问题最合适的方法.

TMVA 分两阶段来完成一个待研究的分类问题. 第一阶段称为训练阶段, 针对使用者提供的同一组训练样本和选定的若干种判别方法, 进行相应的各分类器的训练、测试和性能评估. 这一阶段的任务由程序 TMVAnalysis (运行宏 TMVAnalysis.C) 完成. 第二阶段称为应用阶段, 利用使用者通过第一阶段的各分类器的性能评估后选定的最佳判别方法, 对待分类的实验数据样本进行判别分类. 这一阶段的任务由程序 TMVApplication (运行宏 TMVApplication.C) 完成. 这两个阶段的程序流程框图见图 8.1.

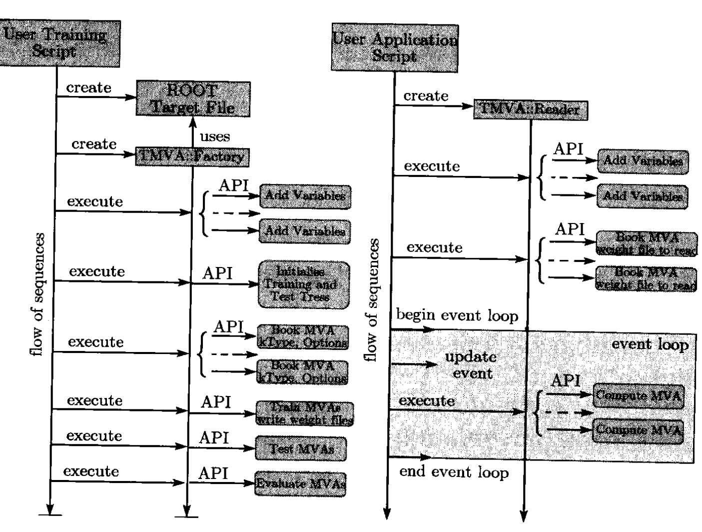


图 8.1 TMVA 训练程序 (左) 和 TMVA 应用程序 (右) 流程框图

在 TMVA 训练程序中, 使用者须提供一个用于训练的脚本文件 (script), 它可以是 ROOT 宏文件, C++ 可执行文件或 python 脚本文件. 该脚本文件用来产生一个 ROOT 目标文件和一个对象文件 TMVA Factory. 后者按照使用者的意愿组织 TMVA 内部程序模块的工作方式. 首先将使用者提供的信号/本底训练样本数据和测试样本数据加以标识和写入内存, 然后以订单的方式 (给定类型标识和自定义的名称) 选择需要测试的判别方法种类. TMVA 按照订单的要求, 逐一对预订的各判别方法进行训练、测试和性能评估. 每种分类器的训练结果写入相应的权文件 (weight file), 而性能评估的诸多直方图写入 ROOT 目标文件. 根据对各分类器的性

能评估数据, 使用者可以选定对所研究问题最合适的判别方法.

在 TMVA 应用程序中, 使用者须提供一个用于应用的脚本文件, 它产生一个 ROOT 目标文件和一个对象文件 TMVA Reader, 后者作为使用者和 TMVA 内部程序模块之间的界面的作用与 TMVA 训练程序中的 TMVA Factory 相仿. 在其初始化阶段, 它写入使用者提供的待测样本的数据, 以订单的方式选定在训练阶段确定的对本问题最合适的分类器, 以及该分类器的权文件 (weight file). 然后 TMVA 逐一读入样本的数据, 并对每个样本的类别用给定的判别方法作出分类判别.

TMVA 还提供了对原始输入数据作预处理的方便手段, 包括对每个事例的贡献作加权处理, 每个输入变量的值变换到 [0,1] 区间内的归一化处理. 归一化处理对于 Fisher 判别, 函数判别分析 FDA, 神经网络判别是必要和有帮助的; 而对决策树林法、多维概率密度估计和 k 近邻法不必要. TMVA 还提供了对原始输入数据作消除线性关联变换的手段. 需要注意的是, 仅对存在线性相关性和高斯分布的输入变量, 消线性关联变换才能发挥其应有的作用, 可改善投影似然比估计、多维概率密度估计、超长方体分割法、决策树林法等分类器的判别性能. 但对于实际的情形, 输入变量往往不满足这些要求. 在这种情形下对原始输入数据作消除线性关联的变换不但无益而且可能有害. TMVA 还提供了对原始输入数据作主成分分析的手段. 一般而言, 主成分分析对于提高分类器的性能是有帮助的.

为了方便初学者熟悉 TMVA 的使用方法和步骤, TMVA 提供了一个练习性的实例来运行宏 TMVAnalysis.C. TMVA 利用一组 TMVA 给定的数据来进行训练和测试. 每个事例的特征向量是线性相关的 4 维正态分布随机变量的样本点. 对于信号事例和本底事例, 4 个分量的期望值和标准偏差各不相同. 训练过程结束后, 提供诸多输出信息, 包括信号/本底事例输入变量的关联矩阵, 各分量在判别分类过程中重要性的次序, 分类器形态参数的总汇, 概率密度的拟合优度 (如果加以申请), 不同分类器判定的信号/本底之间的关联, 信号/本底决策的重叠, 给定本底排除率下的信号效率, 以及分类器其他性能参数的估计, 等等.

训练过程结束后, 除了权文件包含了选定的分类器的训练结果外, 一个使用者图像界面 (graphical user interface, GUI) 被显示出来, 如图 8.2 所示. 它共有 19 个按钮. 按动任一个按钮就可执行相应的 ROOT 宏命令. 图中, (1a)~(1c) 显示信号/本底训练样本的输入变量的原始分布, 退关联变换后的分布和主成分分析后的分布. (2a)~(2c) 显示这三种情形下信号/本底训练样本的所有的一对输入变量的散点图. (3) 显示信号/本底训练样本输入变量间的线性关联系数. (4a) 显示被训练的分类器对于测试样本集的信号/本底分布. (4b),(4c) 相应的概率分布和 Rarity 分布. (5a) 显示分类器的信号/本底判选效率以及信号纯度 (假定信号/本底训练样本事例数相等) 作为分类器判别信号/本底的阈值的关系曲线. (5b) 显示本底排除率 —— 信号效率曲线. (6) 投影似然比判别法中使用的信号/本底概率密度与训练样本数据

的比较. (7a) 多层前向神经网络的各层连接权矩阵. (7b) 多层前向神经网络对于训练样本集和测试样本集的误差参数的收敛性 (检查过度训练). (8) 决策树林法中第一棵决策树的构架图. (9) 分类器的概率密度与训练数据的比较. (10) RuleFit 分类器两维图. (11) 退出图像界面.

(1a) Input Variables
(1b) [Decorrelated Input Variables]
(1c) [PCA-transformed Input Variables]
(2a) Input Variable Correlations (scatter profiles)
(2b) [Decorrelated Input Variable Correlations (scatter profiles)]
(2c) [PCA-transformed Input Variable Correlations (scatter profiles)]
(3) Input Variable Correlation Coefficients
(4a) Classifier Output Distributions
(4b) Classifier Probability Distributions
(4c) Classifier Rarity Distributions
(5a) Classifier Cut Efficiencies
(5b) Classifier Background Rejection vs Signal Efficiency (ROC curve)
(6) [Likelihood Reference Distributiuons]
(7a) [Network Architecture]
(7b) [Network Convergence Test]
(8) [Decision Tree (#1)]
(9) PDFs of Classifiers
(10) [Rule Ensemble Importance Plots]
(11)Quit

图 8.2 TMVA 中用来执行宏命令, 显示训练结果的图像界面 (GUI)

图 8.3~8.6 是练习性实例的一些相关的输出. 如图 8.3 是输入变量 var1~var4 的一些分布. 图 8.4 是输入变量 var3 和 var4 的相互关联. 左图是原始散点图, 可以看到变量 var3,var4 之间存在正关联. 右图是输入变量作了退关联处理之后的散点图, 退关联后的新变量 var3,var4 之间基本消除了相互间的关联.

图 8.5 是 4 种分类器 (投影似然比估计、多维概率密度估计、多层前向神经网络、决策树林法) 对同一组测试样本的输出值 y 的分布. 测试样本分成信号样本和本底样本两类, 对于这两类样本, 对应的 y 的分布是归一化的, 即直方图下的面积等于 1. 因此直方图接近于 y 的概率密度. 按照 TMVA 的设计和约定, 分类器对信号样本的输出值 y 集中于高端 (接近 1), 而本底样本的输出值 y 集中于低端 (接近 0). 当选定一个阈值 $y_{\rm th}$ 作为分类器对 "信号" 和 "本底" 的区别界限, 分类器将 $y > y_{\rm th}$ 的样本判定为 "信号" 事例, 而将 $y < y_{\rm th}$ 的样本判定为 "本底" 事例. 于是, 图 8.5 中, $y > y_{\rm th}$ 的信号样本直方图的面积就是分类器将信号样本判选为 "信号" 事例的判选效率, 用 $\varepsilon_{\rm SS}$ 表示; $y < y_{\rm th}$ 的信号样本直方图的面积就是分类器将

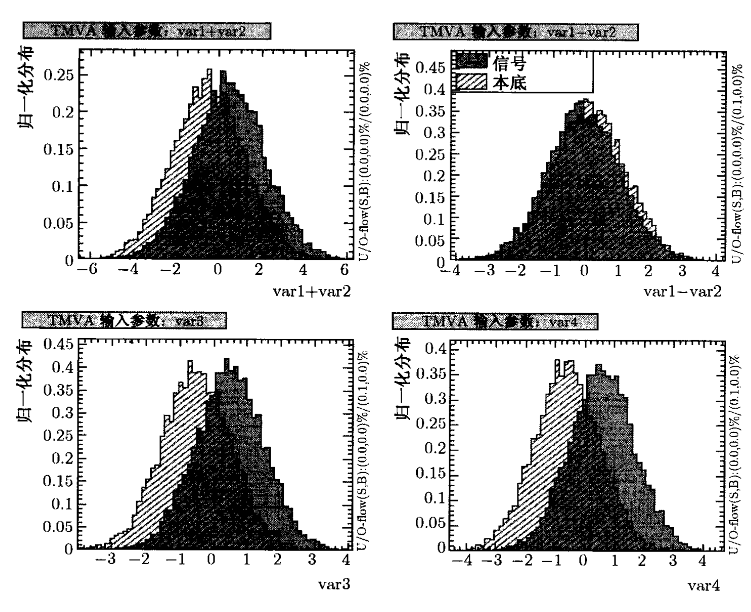


图 8.3　TMVA 练习性实例输入变量 var1~var4 的一些分布

每张图右边的数字表示信号 (S) 和本底 (B) 事例的下溢 (U) 和上溢 (O) 事例数占全部事例数的比例

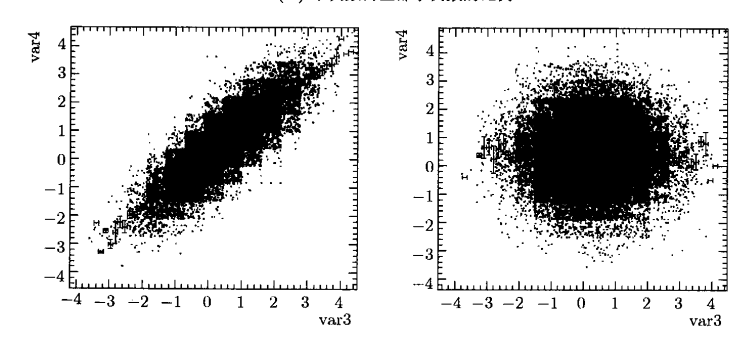


图 8.4　TMVA 练习性实例输入变量 var3 和 var4 的相互关联

左图是原始散点图, 变量 var3,var4 之间存在正关联. 右图是输入变量作了退关联处理之后的散点图, 退关联后的新变量 var3,var4 之间基本消除了相互间的关联

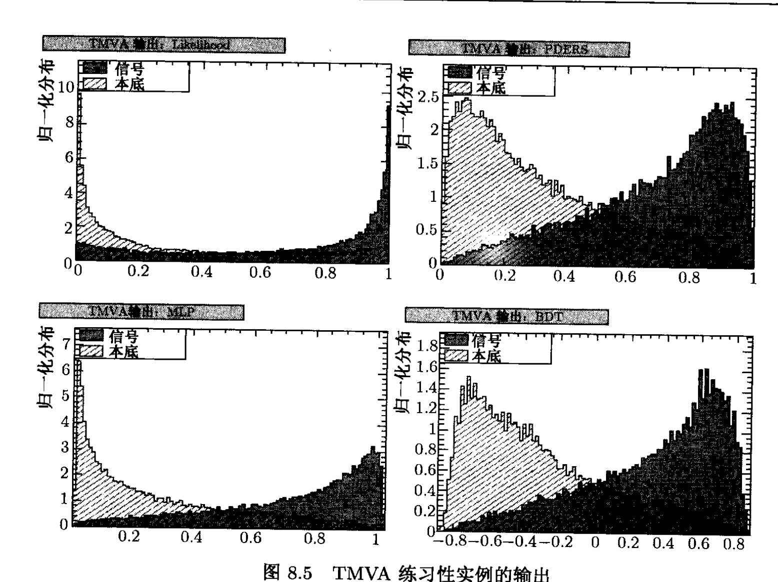


图 8.5 TMVA 练习性实例的输出

4 种分类器对同一组输入样本的输出值分布. 横坐标表示分类器的输出值 y, 纵坐标表示输出值的概率分布. 划斜线的直方图表示对本底样本的输出值分布. 带阴影的直方图表示对信号样本的输出值分布. Likelihood, PDERS, MLP, BDT 分布表示投影似然比估计、多维概率密度估计、多层前向神经网络、决策树林法的判选结果. 每张图右边的数字的含义见图 8.3 的说明

信号样本判选为"本底"事例的概率, 用 $\varepsilon_{\rm BS}$ 表示. 划斜线的 (本底样本) 直方图中 $y < y_{\rm th}$ 那部分面积称为本底排除率 $\varepsilon_{\rm BB}$, 它表示分类器将这部分本底样本判别为"本底"事例从而从"信号"事例中排除出去; $y > y_{\rm th}$ 那部分面积称为本底误判率 $\varepsilon_{\rm SB}$, 它表示分类器将这部分本底样本错误地判别为"信号"事例. 显然有 $\varepsilon_{\rm BB} = 1 - \varepsilon_{\rm SB}$. 一个性能优良的分类器要求信号效率 $\varepsilon_{\rm SS}$ 和本底排除率 $\varepsilon_{\rm BB}$ 同时接近 1, 即分类器将信号样本判为"信号"事例和将本底样本判为"本底"事例的概率同时接近 1.

图 8.6 中的曲线为 5 种分类器对于这组测试样本的信号效率与本底排除率的关系曲线. 由图可见, 对于同样的信号效率, 本底排除率从高到低的次序为多层前向神经网络 (MLP)、决策树林法 (BDT)、多维概率密度估计 (PDERS)、投影似然比估计 (Likelihood) 和 Fisher 线性判别. 类似地, 对于同样的本底排除率, 信号效率从高到低的次序与上述相同. 也就是说, 对于这一特定的测试样本集, 这一次序就是这几种分类器性能优良度的排列顺序 (不考虑计算时间和训练复杂性的因素).

除了给出输出值 y, TMVA 还给出了信号和本底的对于 y 的概率密度 $f_{\rm S}(y)$ 和

$f_{\mathrm{B}}(y)$ (例如图 8.5 所示. 可通过按图 8.2 的 (4b) 按键实现). 依据它们, 可以计算单个样本的分类概率. 样本 i 被分类器判别为 "信号" 事例的概率为

$$P_{\mathrm{S}}(i)=\frac{r_{\mathrm{S}}f_{\mathrm{S}}(i)}{r_{\mathrm{S}}f_{\mathrm{S}}(i)+(1-r_{\mathrm{S}})f_{\mathrm{B}}(i)} \tag{8.2.1}$$

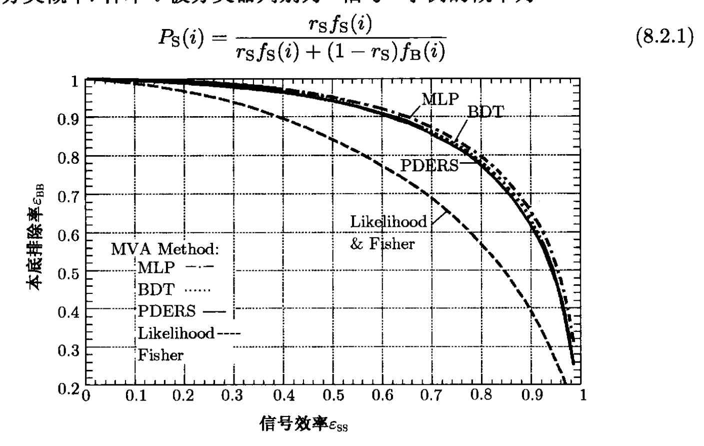


图 8.6 TMVA 练习性实例的输出

5 种分类器对同一组输入样本的输出值分布. 图中曲线为 5 种分类器的信号效率 $\varepsilon_{\mathrm{SS}}$(横坐标) 与本底排除率 $\varepsilon_{\mathrm{BB}}$(纵坐标) 的关系曲线. Fisher 表示 Fisher 线性判别的曲线, 它与投影似然比估计 (Likelihood) 的曲线重合

其中, $r_{\mathrm{S}}=N_{\mathrm{S}}/(N_{\mathrm{S}}+N_{\mathrm{B}})$ 是待分类样本集中信号样本所占比例的期望值; $N_{\mathrm{S(B)}}$ 是待分类样本集中信号 (本底) 样本数的期望值. 依据 $f_{\mathrm{B}}(y)$, 还可计算分类器的 Rarity $R(y_{\mathrm{B}})$ (图像界面 (4c) 按键), 定义为

$$R(y_{\mathrm{B}})=\int_{-\infty}^{y_{\mathrm{B}}}f_{\mathrm{B}}(y)\mathrm{d}y. \tag{8.2.2}$$

其中, $f_{\mathrm{B}}(y)$ 为分类器对于本底样本输出值 y 的概率密度. $R(y_{\mathrm{B}})$ 有如下性质: 对于本底样本, $R(y_{\mathrm{B}})$ 服从 [0,1] 区间的均匀分布. 而对于信号样本, Rarity $R_{\mathrm{S}}(y_{\mathrm{B}})$ 由下式表示:

$$R_{\mathrm{S}}(y_{\mathrm{B}})=\int_{-\infty}^{y_{\mathrm{B}}}f_{\mathrm{S}}(y)\mathrm{d}y. \tag{8.2.3}$$

其中, $f_{\mathrm{S}}(y)$ 为分类器对于信号样本输出值 y 的概率密度. $R_{\mathrm{S}}(y_{\mathrm{B}})$ 集中于 1 附近. 于是可以比较不同分类器的信号 $R_{\mathrm{S}}(y_{\mathrm{B}})$ 分布, 越集中于 1 分类器的性能越好. 投影似然比估计和 Fisher 估计的 Rarity 分布见图 8.7.

可以看到, 两种分类器的本底样本的 Rarity 分布都是均匀的, 但 Fisher 线性判别的信号样本的 Rarity 分布更集中于 1, 因而对于 TMVA 练习性实例的输入数据而言, 它比投影似然比估计有更好的判别性能.

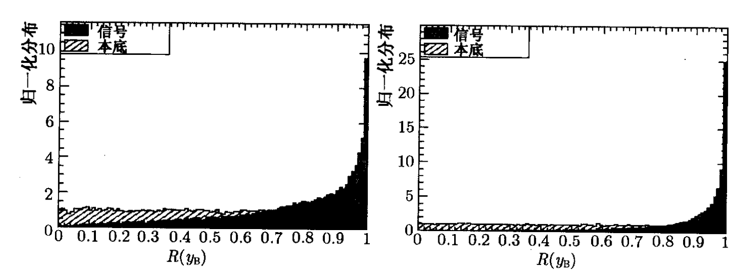


图 8.7 TMVA 练习性实例的输出

投影似然比估计 (左) 和 Fisher 线性判别 (右) 的 Rarity 分布. 划斜线的直方图表示本底样本的 $R(y_B)$ 分布, 即横坐标为 $R(y_B)$ 值, 纵坐标为取值 $R(y_B)$ 的概率; 集中于 1 附近的直方图表示信号样本的 $R_S(y_B)$ 分布. 即纵坐标为取值 $R_S(y_B)$ 的概率. 每张图右边的数字的含义见图 8.3 的说明

所谓的最优分类器很大程度上取决于使用者对于问题的要求. 除了上述的分类器性能图, TMVA 还给出其他一些表征分类器性能的参数供使用者考虑: 3 种具有代表性的本底误判率值 ε_{SB} (等于 1− 本底排除率) 对应的信号效率值 ε_{SS}; 分类器的判别能力 (separation), 定义为[56]

$$\langle S^2\rangle = \frac{1}{2}\int \frac{(f_S(y)-f_B(y))^2}{f_S(y)+f_B(y)}\mathrm{d}y. \tag{8.2.4}$$

式中, $f_S(y)$ 和 $f_B(y)$ 分别是分类器对信号和本底样本的输出值 y 的概率密度 (当 $f_S(y)=f_B(y)$, $\langle S^2\rangle=0$; 当 $f_S(y)$ 与 $f_B(y)$ 相互隔离, $\langle S^2\rangle=1$.).

TMVA 还提供了若干手段, 通过比较训练样本集和测试样本集的同一分类器的分类结果来确定过度训练的影响. 这种比较对于可能存在过度训练的决策树林法和神经网络判别法是有帮助的.

有些分类器的构建过程中需要使用参数拟合方法来求得估计量的最优值, 例如在超长方体分割法、二元决策树中阈值的优化, 函数判别分析中判别函数的优化, 等. TMVA 提供了 4 种拟合程序包可以在 TMVA 环境下使用.

综上所述, TMVA 不但提供了相当多种类的分类器, 而且提供了设计和运行分类器所需的很多相关的功能程序, 它们以一种友好的界面和方式提供给使用者, 因此, 对于实验数据分析工作者是一个极有帮助的多元统计分析系统.

参考文献

[1] T Hastie, et al. The Elements of Statistical Learning. Springer Series in Statistics. Springer 2001.

[2] A Webb. Statistical Pattern Recognition, 2nd Edition. New York: John Wiley & Sons Ltd, 2002.

[3] L I Kuncheva. Combining Pattern Classifiers. New York: John Wiley & Sons Ltd, 2004.

[4] S Haykin. Neural Networks: A Comprehensive Foundation. Prentice Hall, 1999.

[5] 边肇祺, 张学工等编著. 模式识别 (第二版). 北京: 清华大学出版社, 2000.

[6] 李金宗. 模式识别导论. 北京: 高等教育出版社, 1994.

[7] 任若恩等. 多元统计数据分析 — 理论、方法、实例. 北京: 国防工业出版社, 1997.

[8] 朱永生. 实验物理中的概率和统计 (第二版). 北京: 科学出版社, 2006.

[9] M G Kendal, A Stuart. The Advanced Theory of Statistics. Vol. 1, 2, 3. London: Charles Griffin & Company Limited, 1963, 1967, 1966.

[10] W Eadie, et al. Statistical methods in experimental physics. Amsterdam-London: North-Holland Publishing Company, 1971.

[11] 范金城, 吴可法. 统计推断导引. 北京: 科学出版社, 2001.

[12] BES Collab. Phys. Rev. D65 (2002) 052004.

[13] 郑志鹏, 朱永生. 北京谱仪正负电子物理. 南宁: 广西科学技术出版社, 1998.

[14] CERN Program Library. Long Writeup W5013, GEANT, Detector Description and Simulation Tool, CERN, Geneva, Switzerland, 1994.

[15] T Sjostrand. PYTHIA 5.7 and JETSET 7.4 Physics and Manual. LU TP 95-20, 1995.

[16] BES Collab. Phys. Rev. D, 2004, 70: 012006.

[17] R A Fisher. The use of multiple measurements in taxonomic problems. Annals of Eugenics. 1936, 7: 179.

[18] BES collaboration. Phys. Lett. B, 2007, 648: 149~155.

[19] L Breiman, et al. Classification and regression trees. California: Waldsworth International Group, Belmont, 1984.

[20] J R Quinlan. Introduction of decision trees. Machine Learning. 1986, 1: 81.

[21] D Bowser-Chao, D L Dzialo. Phys. Rev. D. 1990, 1993: 47.
M Mjahed. Nucl. Instr. Meth. A. 2002, 481: 601.
R Quiller. SLAC-TN-03-019, 2003.

[22] B P Roe, et al. Boosted decision trees, an Alternative to Artificial Neural Networks. Nucl. Instr. Meth. A. 2005, 543: 577, Physics/0408124, 2004.

[23] Y Freund, R E Schapire. J. Computer and System Sciences. 1997, 55: 119.
L Breiman. The Annals of Statistics, 1998, 26: 801.
R E Schapire, et al. The Annals of Statistics, 1998, 26: 1651.

[24] J R Quilan. Int. J. Man-Machine Studies, 1987, 27: 221.

[25] S Haykin. Neural Networks: A comprehensive foundation. New Jersey: Prentice Hall, 1999.

[26] 袁曾任. 人工神经元网络及其应用. 北京: 清华大学出版社, 1999.

[27] 王永骥, 涂健. 神经元网络控制. 北京: 机械工业出版社, 1998.

[28] 罗四维. 人工神经网络建造. 北京: 中国铁道出版社, 1998.

[29] 王伟. 人工神经网络原理 —— 入门与应用. 北京: 北京航天航空大学出版社, 1995.

[30] C Peterson. Track finding with neural networks. Nucl. Instr. Meth. A. 1989, 279: 537.
L Lonnblad, et al. Finding gluon jets with a neural trigger. Phys. Rev. Lett, 1990, 65: 1321.
D Cutts, et al. Neural networks for event filtering at D0. Comput. Phys. Commun, 1989, 57: 478.
D Cutts, et al. The use of Neural networks in the D0 data acquisition system. IEEE Trans. Nucl. Sci., 1989, 36: 1490.
B H Denby, et al. Neural networks for triggering. IEEE Trans. Nucl. Sci., 1900, 37: 248.

[31] W S McCulloch, W Pitts. A logical calculus of the ideas immanent in nervous activity, Bulletin of Mathematical Biophysics, 1943, 5: 115.

[32] F Rosenblatt. The perceptron: A preceiviing and recognizing automation. Cornell Aeronautical Laboratory Report, 1957, 85-460-1.

[33] D Rumelhart, J McCelland. Parallel distributed processing. Cambridge Bradford Books, MIT Press, 1986, 1: 2.

[34] J Hopfield. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, 1982, 79: 2554.

[35] J Hopfield. Neurons with graded response have collective computational properties like of two state neurons, Proceedings of the National Academy of Sciences, 1984, 781: 3088.

[36] N Metropolis, et al. Equation of state calculations for fast computing machines. Journal of Chemical Physics, 1953, 6: 1087.

[37] G Hinton, et al. Boltzmann machines: Constraint satisfaction networks that learn. Carnegie-Mellon University, Department of computer science technical report, 1984, CMU-CS-84-119.

[38] 徐雷. 一种改进的模拟退火优化组合法. 信息与控制, 1990, 3: 1.

[39] IHEP-BEPCII-SB-13. Preliminary Design Report. The BESIII Detector, Jan. 2004, Institute of High Energy Physics, Beijing 100049, China.

[40] 秦纲等. 中国物理 C (Chinese Physics C), 2008, 32: 1.
秦纲. 北京谱仪 III 粒子鉴别与电磁量能器性能研究. 中国科学院研究生院博士学位论文, 2007, 4.

[41] T M Cover, P E Hart. Nearest neighbor pattern classification. IEEE Trans. on Information Theory, IT-13, 1967: 21~27.

[42] T M Cover. Estimation by the nearest neighbor rule. IEEE Trans. on Information Theory, IT-14, 1968: 50~55.

[43] T Carli, B Koblitz. Nucl. Instr. Meth. A. 2003, 501: 576.

[44] D W Scott. Multivariate density estimation. Theory, Practice, and Visualization, New York: Wiley-Interscience, 1992.

[45] A Hocker, et al. TMVA: Toolkit for Multivariate Data Analysis with ROOT. arXiv physics/0703039; CERN-OPEN-2007-007, 2007; http://tmva.sf.net.

[46] R Sedgewick. Algrithms in C++. Addison Wesley, charpter 26, Boston, USA, 1992.

[47] P C Mahalanobis. Proc. Nat. Inst. Sci. India, Part2A, 1936: 49.
P C Mahalanobis. Proc. Nat. Inst. Sci., Calcutta, 1936, 12: 49.

[48] CERN Program Library. Long Writeup D506. MINUIT, Function Minimization and Error Analysis, CERN, Geneva, Switzerland, 1994.

[49] V Vapnik, A Lerner. Automation and Remote Control, 1963, 24: 774.

[50] V Vapnik, A Chervonenkis. A note on one class of perceptrons. Automation and Remote Control, 1964: 25.

[51] C Cortes, V Vapnik. Support vector networks. Machine Learning, 1995, 20: 273.

[52] V Vapnik. The Nature of Statistical Learning Theory. New York: Springer Verlag, 1995.

[53] B E Boser, I M Guyon and V N Vapnik. A training algorithm for optimal margin classifiers. in D. Haussler, ed., Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, 144, ACM Press, 1992.

[54] R Fletcher. Practical Methods of Optimization. 2nd edition, New York: John Wiley and Sons, Inc., 1987.

[55] C J C Burges. Data Mining and know ledge Discovery, 1998, 2: 121.

[56] The BaBar Physics book, BaBar Collaboration. edited by P F Harrison and H Quinn et al. SLAC-R-0504 (1998); S. Versille, PhD Thesis at LPNHE, http://lpnhe-babar.in2p3.fr/theses/these/SophieVersille.ps.gz, 1998.

[57] I Narsky. StattPatternRecognition: A C++ Package for Statistical Analysis of High Energy Physics Data. arXiv physics/0507143, 2005.

[58] The following web pages give information on available statistical tools in HEP and other areas of science: https://plone4.fnal.gov:4430/P0/phystat/. http://astrostatistics.psu.edu/statcodes/.

[59] R Brun et al. ROOT: An Object Oriented Data Analysis Framework. User Guide 4.04, June 2005.